National Library of Canada
Canadian Theses Service
Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Service des thèses canadiennes

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents

NL-339 (r. 88/04) c

Canada

National Library
of Canada

Canadian Theses Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale
du Canada

Service des thèses canadiennes

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-59189-7

Canada

GRAPHICALLY-BASED AUTOMATED COMPUTER-AIDED MODELLING AND ANALYSIS OF VEHICLE SYSTEMS

Dan Negrin

A Thesis

in

The Department

of

Mechanical Engineering

Presented in Partial Fulfilment of the Requirements
for the Degree of Master of Engineering at
Concordia University
Montreal, Quebec, Canada

July 1990

© D. Negrin, 1990

ABSTRACT

GRAPHICALLY-BASED AUTOMATED COMPUTER-AIDED MODELLING AND ANALYSIS OF VEHICLE SYSTEMS

Dan Negrin

A computer-aided system for the modelling and simulation of physical systems is developed, based on the bond graph formalism. A software package called CANVAS (Computer Analysis of Vehicle Active Suspensions) has been written, enabling a user to create models by interconnecting pre-defined physical system components on a graphical screen. These components (e.g. rigid bodies, springs and dampers, actuators, valves, control system blocks) may be from different physical domains: mechanical, hydraulic, control. Thus, the user may construct relatively complex models from mixed physical domains and may include control system blocks to model feedback control. The software derives a bond graph / block-diagram representation from the input schematic, through a component-by-component translation. The resulting system graph is then causally augmented and processed to produce a set of state-variable dynamical equations in symbolic form. The equations may include nonlinearities either stemming from the components present in the original schematic, or added in by the user. An interactive, graphical post-processing stage is used to solve the equations numerically, and produce output in the form of time histories. Validation of the software is carried out in three phases, through application to several example models, and through correlation with results from the published literature. It is found that the functionality of CANVAS surpasses that of a well-known bond graph-based simulation package, in transcending the primitive bond graph language, and allowing the user to construct models using an easy-to-interpret schematic as input. Confidence in the numerical code is obtained by verifying CANVAS results against analytically known solutions. Finally, it is also found that CANVAS simulation results for models parallelling those in the literature correlate well with the published results.

ACKNOWLEDGEMENTS

The author wishes to express his appreciation and gratitude to Dr. S. Sankar for the encouragement and support extended during the preparation of this thesis.

He would also like to thank all CONCAVE research personnel and other graduate students for their help.

The financial support from CONCAVE Research Centre is also gratefully acknowledged.

Finally, thanks to Susan MacNeil, whose companionship and love have brightened my life, and whose sharp eyes found the 'invisible' errors.

Montreal, Canada
July, 1990

Dan Negrin

TABLE OF CONTENTS

Page

LIST OF FIGURES. ix

LIST OF TABLES xii

NOMENCLATURE xiii

CHAPTER

1. INTRODUCTION. 1

1.1. General 1

1.2. Bond Graph Technique in Systems Modelling . . 2

1.2.1. Historical Perspective 2

1.2.2. Bond Graph Applications. 5

1.2.3. Bond Graph-Based Software. 7

1.3. Specialized Codes 10

1.3.1. Fluid Power Systems. 10

1.3.2. Mechanical Systems 12

1.4. Objective of This Work. 14

1.4.1. Rationale for Developing CANVAS Using Bond Graphs 14

2. MODELLING PHILOSOPHY: COMPONENT MODELLING AND IMPLEMENTATION. 18

2.1. Modelling Philosophy. 18

2.2. Introduction to Bond Graph Terminology. . . . 19

2.3. Component Models. 25

2.3.1. Preliminaries. 25

2.3.2. Conditional Expansion. 26

2.4. Implementation. 27

2.4.1. Bond Graph Elements. 28

Page

2.4.2. Mechanical System Components 33

2.4.3. Hydraulic System Components. 42

2.4.4. Transducers. 55

2.4.5. Block Diagram Elements 56

2.5. Summary 61

3. SOFTWARE OVERVIEW 62

3.1. Introduction. 62

3.2. PRE - Schematic Building. 66

3.3. TOBOND - Conversion of Schematic to Bond Graph 75

3.3.1. System Bond Graph Creation 75

3.3.2. Representation of System Graph 75

3.3.3. Stage I - Component Expansion. 76

3.3.4. Stage II - Component Connection. . . . 76

3.3.5. Bond Graph Simplification. 76

3.3.6. Causality Assignment 77

3.4. BOND_BAS - Automatic Equation Generation. . . 81

3.5. TOSIM - Interface to TUTSIM 83

3.6. POST - Numerical Integration Post-Processor . 84

3.7. Summary 89

4. SOFTWARE VALIDATION AND APPLICATION 90

4.1. Introduction. 90

4.2. Sample Models 92

4.2.1. Radar Pedestal Positioning Servo . . . 93

4.2.2. One-Degree-of-Freedom Oscillator . . . 98

4.2.3. Hydro-Gas Strut. 101

4.2.4. Open-Loop Fluid Power System 105

Page

4.3. Application and Correlation with Published Results 110

4.3.1. Formula One Race Car Active Suspension. 110

4.3.2. Hydraulic Suspension for Crane Vehicle 124

4.3.3. Linear Optimal Active Suspension . . . 138

4.4. Summary 144

5. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK . . . 145

5.1. Conclusions 145

5.2. Recommendations for Future Work 147

REFERENCES 153

APPENDIX 1 157

LIST OF FIGURES

Figure Page

2.1. Schematic of a Physical System and its Representation as a Word Bond Graph and Bond Graph 20

2.2. M1 - One DOF Mass Component 34

2.3. M2 - Two DOF Rigid-Body Component 34

2.4. SPA - Linear Translational Spring 39

2.5. DMA - Linear Translational Damper 39

2.6. SD - Linear Translational Spring / Damper 41

2.7. ACC - Hydro-Gas Accumulator 43

2.8. FVL - Flow Restrictor 46

2.9. RLV - Relief Valve. 46

2.10. CV1 - Three-Port Control Valve. 49

2.11. CY1 - Single-Acting Hydraulic Cylinder. 51

2.12. CY2 - Double-Acting Hydraulic Cylinder. 53

2.13. CY3 - Three-Port Hydraulic Cylinder 54

2.14. CANVAS Transducers. 55

2.15. Example Illustrating Logic Blocks 60

3.1. Stages in CANVAS Model Development. 64

3.2. Flowchart of CANVAS Modules 65

3.3. Mechanical and Hydraulic Components in CANVAS . . 67

3.4. "Black-Box" Representation of Vehicle Suspensions 68

3.5. Ports Display in CANVAS PRE-Processor 73

3.6. "Simplify" Procedure Applied to Sample Bond Graph 78

3.7. Directed Acyclic Graph. 82

4.1. Schematic of Radar Pedestal Positioning Unit. . . 94

4.2. ENPORT-7 Bond Graph for Radar Pedestal Unit . . . 95

Figure Page

4.3. CANVAS Bond Graph for Radar Pedestal Unit 95

4.4. Response of Radar Pedestal Unit Computed by ENPORT. 97

4.5. Response of Radar Pedestal Unit Computed by CANVAS. 97

4.6. CANVAS Schematic of One DOF Oscillator, with Equations. 99

4.7. Displacement Response of 1 DOF Oscillator for Varying Damping Parameter, B. 99

4.8. CANVAS Schematic of Hydro-Gas Strut 102

4.9. Response of Hydro-Gas Strut for Non-Zero Initial Momentum 104

4.10. Phase-Plane Plot of Hydro-Gas Strut Response. . . 104

4.11. CANVAS Schematic of Open-Loop Control System. . . 106

4.12. Response of Open-Loop Control System for Initial Condition: Momentum = 50 Kg·m/s 106

4.13. Response of Open-Loop System to Valve Stroke Input. 109

4.14. CANVAS Schematic of "Formula One" Active Suspension 112

4.15. Hydro-Gas Spring Characteristic 115

4.16. Aerodynamic Input for "Formula One" Example . . . 117

4.17. Displacement Response of "stiff" Active Suspension. 117

4.18. Displacement Response of "soft" Active Suspension. 118

4.19. Frequency Response of "Formula One" Active Suspension 118

4.20. CANVAS Equation Set for "Formula One" Active Suspension 121

4.21. Comparison of Passive and Active Suspension Response to Half-Sine Input. 123

Figure | Page

4.22. Two-Body (4DOF) Roll-Plane Model of Crane Vehicle with Independent-Cylinder Suspension 126

4.23. Two-Body (4DOF) Roll-Plane Model of Crane Vehicle with Linked-Cylinder Suspension. . . . 126

4.24. Bond Graph Structure Derived by CANVAS for the 4 DOF Linked-Cylinder Suspension Crane Model 127

4.25. Simplified Schematic of Independent-Cylinder Model. 128

4.26. Simplified Schematic for Linked-Cylinder Model 128

4.27. Asymmetrical Road Velocity Input for Crane Vehicle. 132

4.28. Roll-Plane Model Used to Obtain Equivalent Cornering Moment 132

4.29. Bounce Response of Independent-Cylinder Model . . 134

4.30. Bounce Response of Linked-Cylinder Model. 134

4.31. Comparison of Independent and Linked-Cylinder Models for Asymmetrical Road Input 135

4.32. Comparison of Cornering Response of Independent vs. Linked Cylinder Suspension for two Area Ratios 137

4.33. CANVAS Schematic for 2 DOF Optimal Linear Active Suspension 139

4.34. Schematic of 2 DOF Optimal Linear Active Suspension (from [45]). 139

4.35. Incomplete State Equations, as Produced by CANVAS. 140

4.36. Completed State Equations, Including State-Feedback. 142

4.37. Step Response of Optimal Linear Active Suspension 143

LIST OF TABLES

Table Page

2.1 Physical System Domains. 21

3.1 Port Compatibility Chart 71

3.2 Components with Arbitrary Number of Ports. 72

4.1 Hydro-Gas Suspension Characteristic. 114

4.2 Feedback Gains Used in [45]. 141

4.3 Validation of Examples 144

NOMENCLATURE

Symbol		Meaning
A	:	effective area
A_h, A_r	:	cylinder head and ram areas, resp.
B	:	viscous damping coefficient
C	:	generalized compliance / capacitance
C_d	:	discharge coefficient
DOF	:	degree of freedom
e	:	generalized effort variable
f	:	generalized flow variable
f_n	:	natural frequency
F	:	force
G	:	gain
H	:	angular momentum
I	:	generalized inertia
J	:	polar moment of inertia
k	:	stiffness
m	:	mass
M	:	moment of a force
n	:	polytropic expansion index
p	:	generalized momentum
P	:	pressure (absolute)
P_0	:	precharge pressure (accumulator)
q	:	flowrate
Q	:	generalized displacement

r	:	cylinder area ratio, A_h/A_r
R	:	generalized resistance
t, t_0	:	time, initial time
v, v_0	:	velocity (instantaneous and initial)
V	:	volume
V_0	:	precharge volume (accumulator)
x	:	linear displacement (instantaneous)
x_0	:	linear displacement (initial)
β	:	bulk modulus
ω	:	angular velocity (absolute)
θ	:	angular displacement (absolute)
ρ	:	fluid density
$\Phi(\cdot)$	:	generalized function

CHAPTER 1

Introduction

1.1 General

The simulation of physical systems using mathematical models is a common task in engineering practice. More specifically, in vehicle engineering, researchers have demonstrated the relative merits of one type of vehicle system design over another by comparing the behaviour of appropriate mathematical models of the different systems. However, the relatively high effort of developing new mathematical models is evidenced by the frequent reutilization of existing models. When a new model is developed, often it is through a manual approach.

Manual approach

The traditional approach to physical systems modelling is a manual, repetitive and sometimes error-prone procedure. Usually it consists of the following steps:

- draw a schematic diagram representing the system under study;
- identify quantities of interest (forces, moments, displacements, etc.);
- write governing equations relating the various quantities and system characteristics; and
- rearrange equations for solution (analytical o numerical).

Once a set of equations exists for the system, various ways can be used to obtain the dynamic response information required:

- expression-oriented simulation language (ACSL [1]*, CSMP, etc.);
- block-diagram simulation language (TUTSIM); and
- purpose-written code calling library routines (e.g. FORTRAN).

The key limitation of these methods is that the most time-consuming step of converting the schematic into equations is performed manually, and must be repeated for each variation in the model. Alternatives to manual equation writing do exist: a number of formalisms have been developed for multibody kinematics and dynamics, as well as for other disciplines. There is one formalism, however, that, as we shall see, spans several disciplines; this is called bond graphs.

1.2 Bond Graph Technique in Systems Modelling

1.2.1 Historical Perspective

Rosenberg and Karnopp [2] popularized the bond graph notation for representing physical system dynamics. First introduced by Paynter [3] in the 1960's, this notation is based on the power exchange that takes place between physical system components. Advantages of the bond graph

* Numbers in square brackets denote references

notation include:

- *Power flow* modelling procedures allow a dynamic description of the system which retains a close association with the physical connections and actions in the modular system itself [4].

- The bond graph technique has turned out to be a great help in modelling complex dynamic systems [5] :

    - Sub-systems may be modelled in detail separately and coupled later on, thereby forming the complete bond graph;

    - Mechanical, hydraulic, electrical or thermal systems are uniformly described using the same technique; and

    - Due to the strict and uniform notation, earlier efforts in modelling particular sub-systems are not lost but can be used by others and inserted in a new bond graph.

- One of the most impressive features of bond graphs, however, is their ability to be transformed, from a rough existential statement of the sort of model of a physical system to be used, to a detailed graph that contains all the information required for the simulation of the system, by means of a series of operations on the original bond graph [6].

● Besides its contribution to the understanding of physical systems, the bond graph method can be very helpful in the preparation of equations for digital simulation. The technique used for this purpose is called causal analysis. By means of causality, implicit equations (e.g. dependent storage elements) can be easily detected [7].

● Although bond graphs in fully augmented causal form are essentially equivalent to signal flow graphs (and hence block diagrams), they are preferable to these because:

- they are more notationally compact; and
- their physical interpretation is more straightforward.

● Bond graphs have the important property of providing explicitly the topological and computational structures of the system. The *topological structure* describes how the components are assembled and the *computational structure* defines the set of mathematical equations describing the system's behavior.

A bond graph in which each bond has its causal stroke is in fact an *algorithm* to calculate the variables in the system and we say that it provides the *computational structure* of the system [8].

The major advantages of bond graph modeling are that in such modeling a topological structure is used to represent the power/energy characteristics of engineering systems, and that systems with diverse energy domains are treated in a unified manner. A topological representation, such as a bond graph, offers great advantage at the conceptual design level, since quantitative details are not required prematurely (*deferred*). In addition, graphical representations document complex models clearly and unambiguously. They often are the easiest way for a group of engineers to communicate the description of energy flows in dynamic systems [8].

Since a bond graph is an unambiguous representation of an energy system, it is possible for a computer program to automatically generate the equations for dynamic analysis of the system [2].

Because the bonds in bond graphs represent the power coupling, such models apply to mechanical, electrical, thermal, hydraulic, magnetic, chemical, and other physical domains. They are especially useful in systems which function in coupled domains, such as electro-mechanical, electro-hydraulic, hydro-mechanical and others.

The major disadvantage of bond graphs is that the notation is new. Experienced modelers sometimes find it difficult to change from the methods of block diagrams and state equations to bond graphs [9].

1.2.2 Bond Graph Applications

Although the bond graph formalism has a limited number of adherents, its flexibility and usefulness should earn it more widespread acceptance. Already, through the efforts of several researchers, bond graphs have been applied to a variety of problems in engineering. The following is a sampling of the areas of application:

- Hydro-pneumatic suspension analysis [10]
- Vehicle Systems [5,11,12]
  - Motorcycle dynamics [13]
  - Heavy Truck [14]
- Lumped-parameter mechanical systems [7]
- Distributed-parameter mechanical systems [15]
- Planar mechanism analysis [16,17]
- Constitutive modelling of nonlinear materials [18]
- Hemodynamics (Physiology) [8]
- Relief-valve dynamics [19]

Felez and Vera [10] used bond graphs to model a crane vehicle in the roll plane with different suspensions. Their work clearly illustrates the modularity of the bond graph approach: starting with the basic vehicle model, each of three suspension subsystems is inserted in the bond graph, yielding the complete model. The suspension models used are: independent cylinder; linked cylinder; and active.

Pacejka and Tol [14] developed bond graph applications in vehicle dynamics including a 3D model of a heavy truck. Noteworthy of this study is the comparison of simulation results to experimental data.

Planar mechanisms are modelled using bond graphs in [16]. The modularity of bond graphs is seen as an advantage in estabilishing a correspondence between the mechanism's physical components and the bond graph. This is seen to lead to automated equation derivation. Zeid [17] also applies bond graphs to the modelling of planar mechanisms: in his work, the equivalence of bond graphs to the Lagrangian approach is estabilished. It should be noted that both works utilize artificial stiffness to describe the mechanism constraint equations.

1.2.3 Bond Graph-Based Software

Several software packages, in varying stages of development have been written to permit systems modelling using bond graphs. The following list includes all the packages known to this author.

THTSIM [20] and TUTSIM [21]
ENPORT-7 [22]
UNISYS [13]
CAMP [23]
MOPS [24] (under development)
POLSYAS [14,19,25]

These programs were developed with various goals in mind, but one characteristic they all share is that they are aimed at the engineer familiar with the bond graph formalism. Because of the limited bond graph user community, this is a major drawback.

A discussion of the capabilities and deficiencies of the state-of-the-art in bond graph-based software is in order.

THTSIM [20] (and TUTSIM [21], the micro-computer version) are block-diagram oriented simulation languages. By providing a variety of block types, this software allows simulation of a large class of continuous and discrete-time systems, including physical plants, control systems and digital hardware. A user of this program creates a structure table, which is a listing of blocks with their inputs. A major disadvantage of working with a structure table is that it becomes the user's responsibility to keep track of such things as block numbers, input lists and parameter values. This makes model construction a tedious task; more importantly, it makes modification even more difficult and error-prone. A further disadvantage of this package is that the user wishing to utilize bond graphs must perform causality assignment manually, prior to entering the structure table. This is a particularly acute inconvenience with large simulation models.

ENPORT-7 [22] is a comprehensive software system used to model physical systems using bond graphs and block diagrams. It includes an interactive bond-graph editor and a graphical output facility. The package is mature, being the first bond graph-based program; it dates back to the 1960's. Its major disadvantage is one it shares with the other packages: the user must be thoroughly familiar with bond graphs in order to use it.

UNISYS [13], performs automatic equation derivation and simulation for nonlinear systems from bond graphs.

CAMP [23] is an interface package between a bond graph model description and continuous system simulation languages such as ACSL or CSMP. It is limited to linear equation generation - any nonlinearity must be included by the user *after* the linear equations are generated. It thus requires manual intervention every time a nonlinear element is present; this is seen as a major drawback.

MOPS [24] is a bond graph preprocessor similar to CAMP; it produces an input file for the continuous systems simulation language CSSL IV. Again, the usefulness of the package is limited by the need to intervene manually to alter the equations when nonlinearities occur.

POLSYAS [14,19,25] (Polydescriptive, Polyalgorithmic Simulation System for Nonlinear Continuous Problems) also generates a set of first-order differential equations suitable for numerical integration. Unfortunately, the equations are written in a very user-unfriendly manner, so that it is difficult to inspect and/or modify them.

As a final note, one should be aware that none of the above-mentioned packages aid the user in constructing the system model in the first place. The highest level of abstraction provided is that of the bond graph element, and this is found to be insufficient due to the following reasons:

- commonly occurring subsystems have to be modelled explicitly each time;
- users familiar with bond graphs are scarce; and
- the equations derived from the input graph are difficult to relate to the original physical system.

1.3 Specialized Codes

One important drawback of the general-purpose programs is that problem formulation can be very tedious. The user is required to perform the transformation from a specialized problem domain with which he is familiar into a suitable description. This burden can vary from having to learn the syntax of a simulation language such as ACSL to having to absorb a whole formalism such as bond graphs for packages based on the latter.

In recognition of this limitation, several specialized codes have been written to cope with complex physical system models in a particular area. Among the engineering disciplines that have such specialized codes are Structural Mechanics (Finite-Element codes), Electrical Engineering (Network Analysis), Fluid Power Systems, Multibody Dynamics and Vibration. An overview of existing software from two of these fields is presented below.

1.3.1 Fluid Power Systems

Fluid power engineering has several specialized codes. Among these: CATSIM [26], HOPSAN [27], the program by Kinoglu et al. [28], and HYSAN [29].

CATSIM [26] is a catalogue-based simulation package for hydraulic systems. Model building is accomplished by using a standard methodology to link together component subroutines written in FORTRAN and stored in a library. Parameter information is automatically requested for each component, according to its structure. Simulation of the complete system is performed using the HOPSAN package [27].

HOPSAN is a general simulation program specially conceived for the simulation of fluid power systems. It requires a model to be developed separately and linked. The program's function is then to allow solution of simulation models and plotting of results. The user has interactive control over the output of the program, and various analysis options are available.

Kinoglu et al. [28] developed a software package to simplify the modelling of fluid power systems. The key elements of their package are a graphical pre-processor to build system schematics, and a post-processor to graphically display simulation results. Simulation is performed using special-purpose fluid power codes found in the public domain (AFSS [30]), so that essentially the pre-processor builds a model representation suitable for these codes.

HYSAN [29] is used for the dynamic simulation of hydraulic components and systems. A graphical editor is used to create a hydraulic system schematic. The package contains a limited number of mechanical elements. It generates equations for numerical integration, but these are 'invisible' to the user. Output is in the form of transient response plots, or a tabulation of pressures, displacements, flows, etc.

1.3.2 Mec:hanical Systems

GMR DYANA [31,32] was one of the earliest reported efforts aimed at developing a user-friendly modelling tool. It was designed primarily to study systems whose mathematical model is identical in structure and form to the model of a spring-mass-damper system. This limitation was later alleviated by extending the program to allow holonomic constraint equations as part of the model. The software was one of the first to include an analytical expression differentiator.

Dix a:d Lehman [33] developed a computer program (MEDUSA) for the numerical solution of machine dynamics problems. Their system performed automatic solution of mechanical systems including rigid links with mass, flexible]inks, springs and dampers, and force and motion generators. They developed a formalism called 'Information Flow Diagram' which is remarkably similar to bond graphs, and which expresses the interaction between system components. The user of the system is expected to manually prepare such a diagram, and then write a special FORTRAN subroutine calling various subroutines in proper order.

In the field of vibration analysis a noteworthy package [34] was developed using an object-oriented programming system (OOPS) based on Smalltalk-80. The package models two-dimensional multibody vibration systems. Input to the system is through an interactive graphic window to define the system model; elements are chosen from a menu that includes rigid bodies, particles, connection points, springs, dampers and various constraint elements. Output is a set of differential equations in symbolic form; these may be written out at any time during model development.

MEDYNA [35] is a powerful but complex system for the modelling and simulation of controlled mechanical systems. It includes special modules that gear it toward rail vehicle dynamics. Its flexibility and power, however, tax the user with a cumbersome model development cycle.

DADS [36] was developed as a mechanical system simulation package. It includes two-dimensional and three-dimensional kinematics and dynamics. Various constraint elements (sliders, revolute joints) as well as the usual springs and dampers allow construction of realistic models. Model development, however is a laborious process due to the text-oriented nature of the preprocessor.

CAMSYD [37] is a package for the analysis of a special class of lumped parameter mechanical systems. These systems are composed of rigid bodies interconnected by springs, dampers and revolute joints. The bodies may undergo small angular motions; the forces generated in springs and dampers can be passive or active, linear or nonlinear. For this class of multi-body systems, CAMSYD offers a mathematical formalism which leads to straightforward symbolic derivation of the system dynamic equations.

1.4 Objective of this Work

An analysis of the deficiencies of existing modelling and simulation software packages reveals that there is a need for a package that combines the usefulness and 'expertise' of the specialized software with the generality and applicability of general-purpose programs/languages.

The major objective in the work carried out is therefore to create a software package to automate the process of model construction, equation derivation and solution - *from schematic to response.*

The approach chosen allows the engineer to describe a model to the computer by interactively assembling its schematic on a graphics screen. The software converts the schematic into an intermediate representation based on bond graphs, then derives state equations; numerical solution of the equations and graphical output of results are the final links in the chain.

The package we developed is entitled CANVAS - Computer ANalysis of Vehicle Active Suspensions - in deference to its intended main area of application.

1.4.1 Rationale for Developing CANVAS Using Bond Graphs

The choice of bond graphs over other formalisms for the implementation of CANVAS software is based on some fundamental advantages that distinguish bond graphs from any other modelling language.

Specialized analysis codes often cannot model physical systems consisting of components from different domains (multi-body mechanical, fluid, electrical, thermal, etc.). Because each of the specialized packages uses its own special formalism for representing the components, interfacing elements from different domains is extremely difficult, if not impossible. In particular, when modelling active suspension systems, components from different energy domains coexist in the same physical system. As aptly put in [35],

> ...the diversity of components acting as coupling elements - they can consist of electrical, hydraulical [sic], pneumatical [sic] as well as magnetic and mechanical parts - rules out an equation generation based on a few axioms as it was done for the rigid and elastic modelling.

As we shall see, bond graphs are just the key for using only "a few axioms" to generate equations.

A central issue in the design of CANVAS was the ability to model such 'mixed' physical systems. This is the primary motivation for the adoption of bond graphs for component modelling. Although developed primarily for the analysis of vehicle active suspensions, the package is indeed a general-purpose modelling and simulation framework based on bond graphs.

Bond graphs make it possible to describe system component models from different energy domains using a small set of symbols that represent power exchange and dissipation and energy storage in physical systems. This makes the modelling of heterogeneous systems a very natural process, as long as they can be approximated by lumped parameter models.

Any physical system which can be modelled using a lumped-parameter state variable formulation is a candidate for modelling using CANVAS. This includes mechanical, hydraulic, electrical and thermal systems - or any combination thereof. The package was developed based on the recognition that it is possible to create mathematical models of physical systems automatically, starting from a schematic diagram. Systems equations are written symbolically, and the system response is given in numeric and graphical form.

Because it provides a meaningful graphical interface, our software package is a convenient and user-friendly tool, rather than a cumbersome computer program that forces the user to learn a new notation. In fact, the task of creating system models and equations is reduced in complexity to the point that it is easy to construct relatively elaborate models and obtain solutions in very little time.

Our approach in the conception of CANVAS has been to provide a *convenient* modelling tool. In fact, CANVAS combines modelling capabilities from fluid power systems to vehicle ride vibration problems, and furthermore enables the construction of arbitrary lumped-parameter models using bond graphs or block diagrams. Although CANVAS is based on a bond graph component model representation, the user of the system hardly needs to know this. All existing bond graph based packages assume that the user is familiar with the bond graph notation.

An important consideration in the design of CANVAS was the recognition that the user does not want to be limited to the use of primitive bond graph symbols when creating a system model. Rather, the user would prefer to work with entities that are representative of the physical system, while leaving the details of bond graph development to the software. This is especially important, because although the bond graph modelling approach is highly regarded by many, it is well known by few.

Many of the features and operational characteristics of the CANVAS package are inspired from the pioneering bond graph software ENPORT. However, we have gone beyond the capabilities of ENPORT in developing our package; most importantly, we insulate the user from the underlying bond graph model representation by providing component models with a user-friendly schematic editor.

CANVAS then, with its graphical pre-processor and its component models, obviates the need for the user to know about bond graphs. Of course, the CANVAS user knowledgeable about bond graphs can benefit from many of the features offered by the other programs; but the intent in developing CANVAS was to use an easy-to-understand graphical user-interface.

CHAPTER 2

Modelling Philosophy: Component Models and Implementation

2.1 Modelling Philosophy

As discussed in the first chapter, there is strong motivation for basing a graphical modelling framework on a bond graph representation. In this chapter, we discuss the modelling philosophy of CANVAS, which uses a *bottom-up* approach; we also introduce enough bond-graph terminology to enable the reader unfamiliar with the notation to understand the remainder of this work.

The CANVAS user creates models by a process of interactive assembly of existing component models. For this reason, the model-building process is dubbed *bottom-up*. By contrast, in a *top-down* approach, the user would start with generic models, and specify details at a later stage.

The advantage of the bottom-up approach is that at any stage, the user has a complete understanding of the model he or she has created. A disadvantage is that a certain amount of commitment to a particular model is inevitable in this approach: once a component is assembled into a model, it becomes an integral part of it.

However, mitigating this disadvantage are two features of CANVAS that will be discussed later: generic components (e.g., force generator) and deferred component model specification.

The modelling philosophy of CANVAS is based on the realization that a great variety of useful physical system models (in general) can be built using a relatively small set of (bond graph) symbols to represent physically meaningful, discrete components. This is the central concept of the thesis.

2.2 Introduction to Bond Graph Terminology

Although a detailed discussion of bond graphs is unnecessary here, a few of the terms that will recur in the thesis are introduced for the reader's convenience. Rosenberg and Karnopp [2] have published an excellent introduction to the field for the more interested reader.

Bond graphs are a graphical representation for mathematical relationships. They describe power exchange between elements of a physical system by means of power bonds. The engineer constructing a bond graph model starts with a conceptual model called 'word bond graph'. In the word bond graph, each discrete component is represented by a word, and components are interconnected by bonds. Fig. 2.1 (taken from [38]) shows an example of a word bond graph associated with a simple schematic.

Bonds represent a bi-directional signal flow, just as signal-flow-graph and block diagram arrows represent a single signal flow. Indeed, as suggested by Karnopp and Rosenberg [6], a bond graph may be transformed into a block diagram in a systematic fashion if desired.

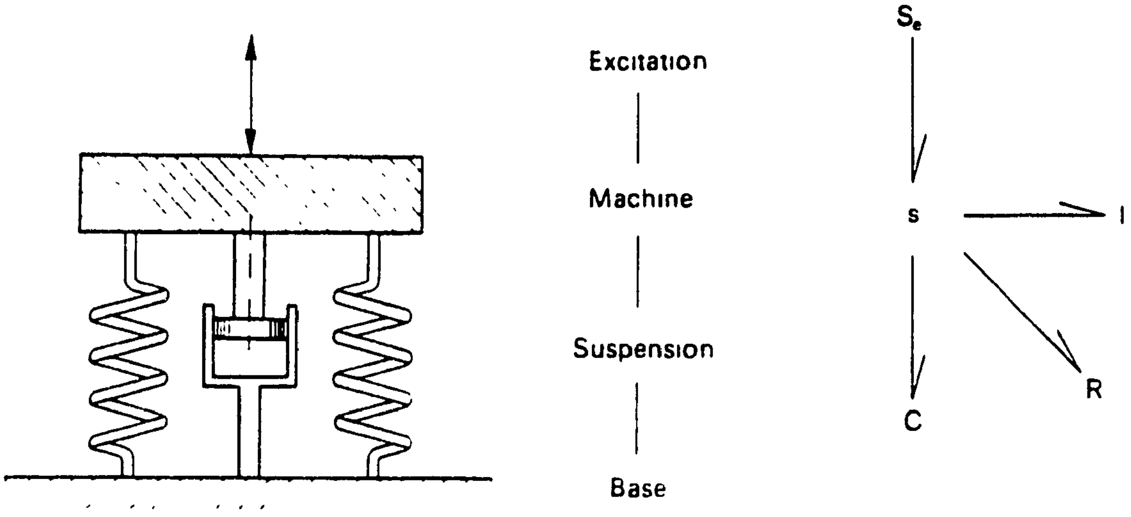


Fig. 2.1. Schematic of a Physical System and its Representation as a Word Bond Graph and Bond Graph

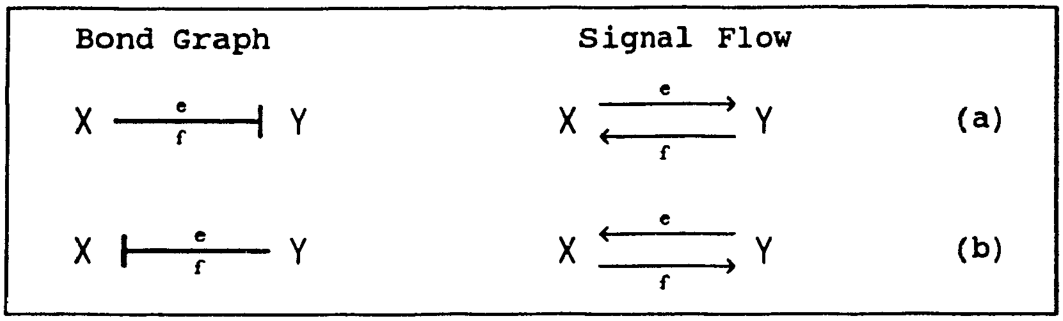
Signal Flow in Bond Graphs

The signals in bond graphs are called *efforts* (e) and *flows* (f). Effort is a generalized term for quantities analogous to force, voltage, or pressure. Flow is a generalization of velocity, current, or flowrate. By providing a unified notation to express relationships amongst these various types of quantities, bond graphs tie together *multi-domain* systems. Table 2.1 lists the physical system domains covered by bond graphs.

TABLE 2.1 - Physical System Domains

	Physical Domain			
	Mechanical	**Hydraulic**	**Electrical**	**Thermal**
Effort, e	Force Torque	Pressure	Voltage	Abs.Temp.
Flow, f	Velocity Ang.Velocity	Flowrate	Current	Entropy Flow
Generalized Displacement Q	Displ. Ang. Displ.	Volume	Charge	Entropy
Generalized Momentum, p	Momentum Ang. Mom.	—	Vol.Pulse	—

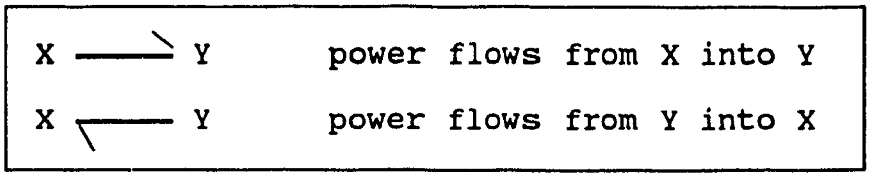
The manner in which signal flows are represented by bond graphs is shown below.



The vertical bar at either end of a power bond is the *causal stroke*. The element next to the causal stroke imposes a flow quantity on the other element, which in turn imposes the effort quantity. Thus in (a), element X is imposing an effort on element Y. In (b), the reverse is true. Causality is a concept of fundamental importance in systems modelling.

Power Direction

Signal flow direction is insufficient to determine the equations from a bond graph. Sign information is necessary for effort or flow balance, and for writing constitutive equations. Such sign information is provided by the power-direction arrow, which is independent of the causal stroke previously discussed. The power arrow indicates the assumed direction of positive power flow between two elements. It is intimately tied with the assumed sign convention for the various flows and efforts.

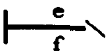
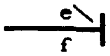
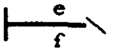
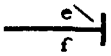


power flows from X into Y

power flows from Y into X

Constitutive Equations

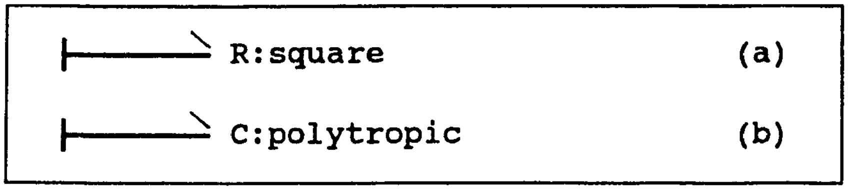
Constitutive equations are expressed by capacitance (C), inertance (I), resistance (R) elements. In generic form, these elements have equations as given below.

 C	Capacitance	$e = \Phi(Q)$
	(integral causality)	$Q = \int f \cdot dt$
 I	Inertance	$f = \Phi(p)$
	(integral causality)	$p = \int e \cdot dt$
 R	Resistance	$e = \Phi(f)$
 R	Conductance	$f = \Phi(e)$

For the resistance, $\Phi(\cdot)$ expresses effort as a function of flow. The validity of either causality hinges on the single-valuedness of $\Phi(\cdot)$.

Constitutive laws are not explicitly expressed by the bond graph language. Rather, the systems modeller places generic C, I and R elements where appropriate in the system bond graph, and then substitutes constitutive laws when preparing the set of equations. A colon placed next to the bond graph element, with a note written next to it indicates that a particular constitutive law is to be assumed when expanding the bond graph into equations. Such bond graph elements are referred to as *annotated* elements.

For instance, (a) below indicates that a square resistive law is to be used, whereas (b) indicates that a polytropic capacitance law is to be substituted.



Additional elements (transformers, junctions and gyrators), termed 'junction structure', are used to define the channeling of energy, as well as its transduction from one domain to another.

Our Approach

The novelty in the work carried out for this thesis resides in the approach to modelling physical systems. In essence, the engineer constructs models of physical systems by merely interconnecting pre-defined model entities. These entities are represented by primitive bond graph and block-diagram symbols; they may be thus be viewed as 'bond graph macros'.

The modularization of system equations is a direct consequence of the use of bond graphs for their representation. The elegance of the approach resides in its open-endedness and uniformity.

In fact, the usual conversion of a 'word bond graph' [6] into a system bond graph is performed automatically in CANVAS.

2.3 Component Models

As previously mentioned, component models in a CANVAS schematic are expanded into bond graph / block diagram structure; from this structure equations are written. The following details the expansion of some of the CANVAS component models and is essentially a description of the behaviour of the TOBOND module. Assumptions, sign conventions and other component-related information are presented here, on a component-by-component basis. Some preliminary facts about the approach to component modelling are also introduced.

2.3.1 Preliminaries

CANVAS components possess a number of attributes that enable the user of the software to work with graphical entities during model building, while leaving the details of equation derivation to the package. These attributes are:

- A clearly recognizable and meaningful graphical representation (icon) for the pre-processor;
- Pre-defined connection points (ports) visible to the user;
- Port compatibility information needed to disallow connections that violate physical sense (e.g. a mechanical to a hydraulic port);
- Model structure information used to create the system model in bond graph form; and

- Conditional expansion information for selected components, as detailed below.

2.3.2 Conditional Expansion

Certain physical system components (e.g. a hydraulic cylinder) are often modelled differently according to the situation. For instance, considering a hydraulic cylinder, a capacitive effect may be included to describe fluid compressibility within an enclosed volume; or piston leakage could be of interest, in which case a leakage resistance may be appended.

Such *conditional modelling* appears frequently in the literature, and it is usually the engineer who decides what component model is appropriate under a given set of circumstances. As noted in [4], "Whether or not the fluid capacitance effect of a particular component should be considered in the dynamic analysis of a system is largely a matter of experience and judgement."

In order to endow CANVAS with this conditional modelling feature, some of the component models are expanded interactively (by module TOBOND.PRO) when a choice of models exists.

Returning to the example of a hydraulic cylinder, physical effects that could be optionally included are:

• Fluid Compressibility	(Compliance)
• Piston Leakage	(Resistance)
• Seal Friction	(Resistance)

The rationale for providing components with conditional expansion is based on the following points:

- Different system models will be obtained by different expansions of the same component;
- Computational difficulties (e.g. numerical stiffness / derivative causality) can be avoided by proper model selection. Conversely, an inappropriate model may give rise to computationally unwieldy models; and
- Automating the model selection by a knowledge-based process becomes possible when models are left partially unspecified. This means that a knowledge-based system can select a starting model configuration, and revise the component model until a satisfactory system model is obtained.

2.4 Implementation

Appendix 1 is a complete listing of existing CANVAS component models. The remainder of the chapter presents a selection of the more important components, makes explicit the underlying assumptions and sign conventions, and details the equations that are automatically written for each. It is not attempted here to describe each and every component model, and this is principally because CANVAS is an extensible package, where it is possible to add new component models.

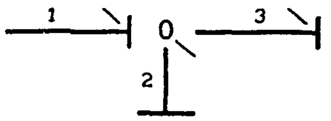
2.4.1 Bond Graph Elements

Although not restricted to modelling systems using primitive bond graph symbols, CANVAS does provide these symbols for the user who is familiar with the notation. In this section we present the building blocks that form the lowest level of abstraction within CANVAS.

0-junction Parameters: -none-

The common-effort junction is a means of simultaneously writing a flow-balance equation, and imposing the constraint of a single effort on all attached bonds. Those bonds which 'point' into the junction are considered to have positive flow, those pointing out, negative flow. Example:

Bondgraph Structure



Equations

$$e_2 = e_3 = e_1$$

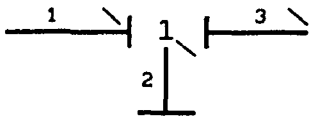
$$f_1 + f_2 - f_3 = 0$$

$$e_1 f_1 + e_2 f_2 - e_3 f_3 = 0$$

1-junction Parameters: -none-

The common-flow junction is a means of simultaneously writing an effort-balance equation and to constrain all attached bonds to a single flow. Example:

Bondgraph Structure



Equations

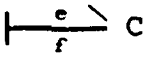
$$f_1 = f_3 = f_2$$

$$e_1 + e_2 - e_3 = 0$$

$$e_1 f_1 + e_2 f_2 - e_3 f_3 = 0$$

C Linear Compliance **Parameters: C, Q_0**

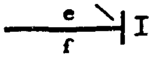
This element has a capacitive constitutive law relating flow and effort. The set of all C-elements in a system bond graph is called C-field; each C-element in integral causality yields one state variable; the term *explicit C-field* refers to an all-integral causality C-field.

Bondgraph Structure	Causality	Equations
	Integral	$e = \frac{Q}{C}$ $Q = Q_0 + \int_{t_0}^{t} f\cdot dt$

e.g.: a linear spring, a tank filled with fluid, and an electrical capacitor.

I **Linear Inertance** Parameters: I, P_0

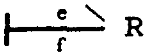
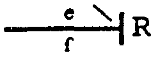
This element has an inertive constitutive law relating flow and effort. The set of all inertance elements in a system bond graph is referred to as the *I-field*. Each I-element in integral causality yields one state variable for the system graph.

Bondgraph Structure	Causality	Equations
	Integral	$f = \frac{p}{I}$ $p = p_0 + \int_{t_0}^{t} e\cdot dt$

e.g.: a moving mass, a rotating wheel, fluid inertance in a pipe, and an electrical coil.

R Linear Resistance Parameter: R

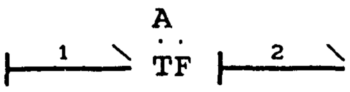
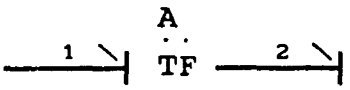
For this element, flow and effort are related according to causality, as shown below.

Bondgraph Structure	Causality	Equations
	Resistance	$e = f \cdot R$
	Conductance	$f = \frac{e}{R}$

e.g.: a mechanical dashpot, hydraulic resistance in pipe flow, and an electrical resistor.

TF, MTF Transformer Parameters: [A]

The transformer element relates two effort-flow pairs, according to causality:

Bondgraph Structure	Equations
	$e_1 = A \cdot e_2$ $f_1 = \frac{1}{A} \cdot f_2$
	$e_2 = \frac{1}{A} \cdot e_1$ $f_2 = A \cdot f_1$

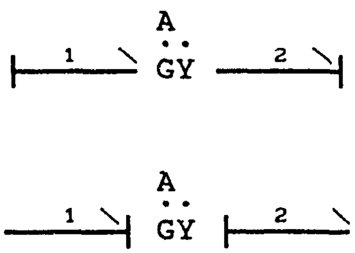
where A is known as the (fixed) transformer modulus.

e.g.: a simple lever, two gears, a piston (positive displacement) pump, a hydraulic cylinder, and an electrical transformer.

A generalization of the transformer is known as the modulated transformer, MTF. For the MTF, the modulus is given by a signal originating somewhere else in the model, and not by the fixed parameter, A. Stated another way, the transformer modulus can vary in time but is not necessarily an explicit function of time.

GY, MGY Gyrator Parameters: [A]

Bondgraph Structure



Equations

$$e_2 = A \cdot f_1$$

$$e_1 = A \cdot f_2$$

$$f_2 = \frac{1}{A} \cdot e_1$$

$$f_1 = \frac{1}{A} \cdot e_2$$

Just as the modulated transformer (MTF) has equations identical to those of the fixed transformer (TF), so the modulated gyrator, MGY element has equations identical to those of the fixed GY element; again the modulus is derived from an external signal.

e.g.: gyroscope, centrifugal pump, and an electrical voice coil.

SE Source of Effort Parameters -none-

This element is used to impose effort variables on other elements. Its symbol and equation are given as

Bondgraph Structure	Equations
$\mathrm{SE}\ \overset{e}{\underset{f}{\rule{2em}{0.4pt}}}\!|$	$e = \Phi(\cdot)$

where the generic function can be an input, or a known function of time.

e.g. an applied force, a constant-pressure power supply, a voltage source.

SF Source of Flow Parameters -none-

This element is the dual of the effort source. It creates flow inputs into a system graph.

Bondgraph Structure	Equations
$\mathrm{SF}\ |\!\overset{e}{\underset{f}{\rule{2em}{0.4pt}}}$	$f = \Phi(\cdot)$

e.g. a velocity profile, a fixed-displacement pur.p, and a current source

MULTIPORTS

Additional bond graph symbols called *multiports* exist. These offer a compact way to simultaneously express multiple relations, in a manner analogous to matrices. Their implementation is, however, very case-specific and requires the user to be thoroughly familiar with bond graphs. For these reasons, multiports are not implemented within CANVAS.

2.4.2 Mechanical System Components

M1 1DOF Lumped Mass **Parameters: m, v_0, x_0**

This component is a translational inertia which contributes 2 state variables (linear momentum and absolute displacement) to the set. Its constitutive equation is a linear form of the general inertive law (Eqn. 2), with m = mass, v_0 = initial velocity and x_0 = initial displacement.

$$p = m \cdot v_0 + \int_{t_0}^{t} F \cdot dt$$

$$v = \frac{p}{m}$$

$$x = x_0 + \int_{t_0}^{t} v \cdot dt$$

This lumped-mass element is used in vehicle studies as a representation of the vehicle body's inertia (sprung mass), as well as for unsprung mass comprised of tires, suspension components, and the like.

Implementation of this equation in bond graphs requires an I-element corresponding to the translational inertia, with an attached 1-junction to represent the absolute velocity and to express the force-balance equation. Fig. 2.2 shows the bond graph structure corresponding to this component. The (optional) attached integrator is used in computing the displacement, x.

The attached SE:mg element is shown to illustrate the effect of attaching a Gravity Force element (GRV) to the M1 rigid body. The GRV element is discussed below.

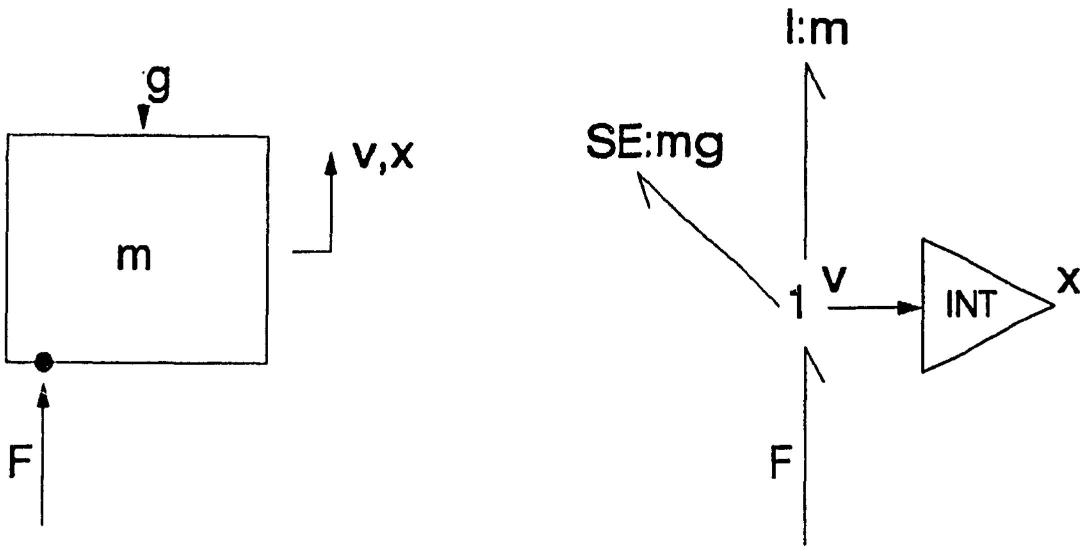


Fig. 2.2. M1 - One DOF Mass Component

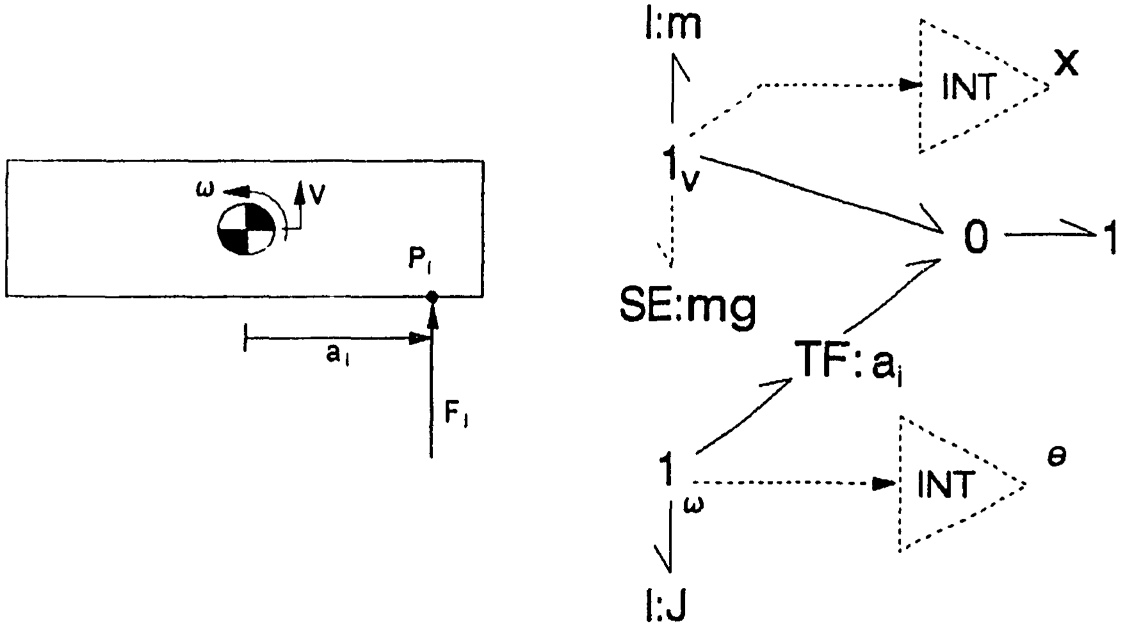


Fig. 2.3. M2 - Two DOF Rigid-Body Component

M2 **2DOF Rigid Body** Parameters: $m, J, v_0, \omega_0\ [, a_i]$

Representing a rigid planar body with translation along one axis and rotation about one axis (passing through the body's center of gravity), this component is commonly used to model vehicle systems in the roll or pitch planes. The translational inertia of the element is given by the mass, while the rotational inertia is given by the polar moment of inertia (about the c.g.).

The algebraic relationship between forces acting on the body at points P_i and moments about the c.g. is

$$M_{c.g.} = \sum a_i \cdot F_i = \frac{dH}{dt}$$

where a_i is the distance from the c.g. to point P_i (positive to the right), and F_i is the applied force (positive upward). Here it is assumed that the body undergoes only small-angle rotations, which is reasonable for vehicle roll and pitch. Refer to Fig. 2.3.

The force balance equation for the element is

$$\Sigma F_i = \frac{dp}{dt}$$

The energic structure for this component is fixed, consisting of I-elements for the rotational and translational inertias; the junction structure, however varies according to the number of force application points, so that for this element a special treatment is required. Each connection point gives rise to junction structure comprised of:

- Two-port Transformer, to implement the equations

$$M_i = a_i \cdot F_i$$

$$u_i = a_i \cdot \omega_{c.g.}$$

- 0-junction, to express the velocity at point P_i as follows:

$$v_i = u_i + v_{c.g.}$$

The 1-junction attached to the rotational I-element provides the moment-balance equation, while the 1-junction attached to the translational inertia expresses the force balance. Again, integrators are used to yield (absolute) angular and translational displacements.

GRV Gravity Force Element **Parameter: g**

This is a special element that can be attached to M1 or M2 rigid body elements. It is a force source that automatically applies a downward (negative) force to the rigid body. In the case of the M2 rigid body, the force is, naturally, applied at the c.g. The parameter, g, is used to specify the gravitational constant, which is dependent on the system of units being used for the remainder of the model.

GRD Ground **Parameters: -none-**

This component defines mechanical ground points, i.e., points constrained to the inertial reference frame. The equivalent bond graph structure is simply a flow source with constant value of zero.

SMO Source of Motion Parameters: -deferred-

When it is desired to impart a given motion to some mechanical part of a CANVAS model, this translational motion source is used to specify a given *velocity* profile. The actual form of the velocity profile is determined at simulation time: any of the source blocks may be substituted in (explaining why the parameters are deferred).

SEF External Force Parameters: -deferred-

Applied forces are treated by this element, which may be attached to mechanical components. An example of usage of external forces appears in Chapter 4, where aerodynamic downforce acting on a Formula 1 car is modelled. As in the SMO component, the actual force time function is specified at simulation time.

MOM External Moment Parameters: -deferred-

Applied moments are useful in predicting the roll behaviour of a vehicle. This component may only be attached to an M2 rigid body. It is then interpreted by the TOBOND module, and is thereby made to act on the 1-junction corresponding to the angular velocity. Again, the actual time function is selected when a simulation is run.

SPA Linear Spring Parameters: C, Q_0

The linear translational spring is characterized by the constitutive law

$$F_s = \frac{Q}{C} \tag{1}$$

$$Q = Q_0 + \int_{t_0}^{t} (V_1 - V_2) \cdot dt$$

where compliance (the inverse of stiffness) has been used.

In integral causality, this component yields one state variable (the compression of the spring). Referring to Fig. 2.4, each endpoint of the spring is represented by a 1-junction: The relative velocity between the endpoints is given by the 0-junction. It is integrated to yield a relative displacement, which in turn yields a force.

DMA Linear Damper Parameter: B

Causality	Equation	
effort-out (resistance)	$F_d = B \cdot (V_1 - V_2)$	(2a)
flow-out (conductance)	$V_1 - V_2 = \frac{F_d}{B}$	(2b)

This component relates a force to the relative velocity of its endpoints. The endpoint velocities are represented by 1-junctions, with a 0-junction in between to take the difference. To the latter is attached the dissipative R-element. Refer to Fig. 2.5.

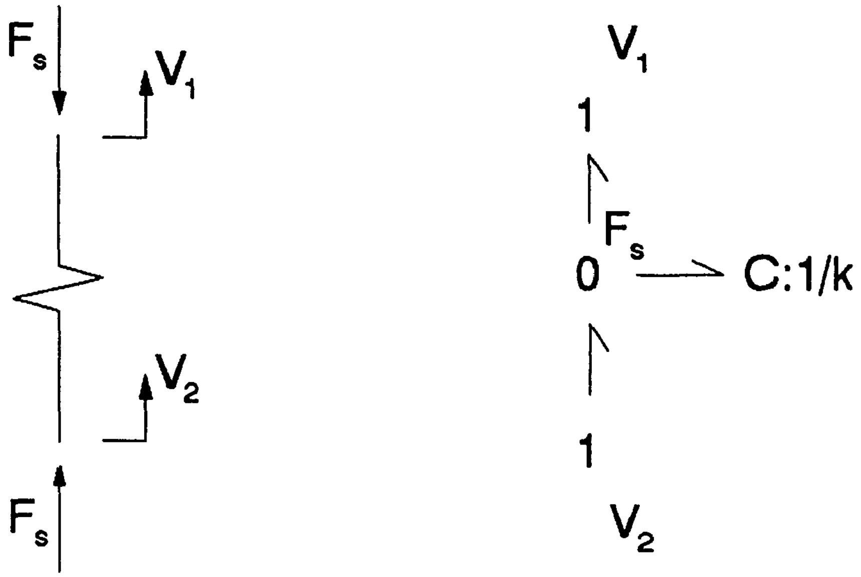


Fig. 2.4. SPA - Linear Translational Spring

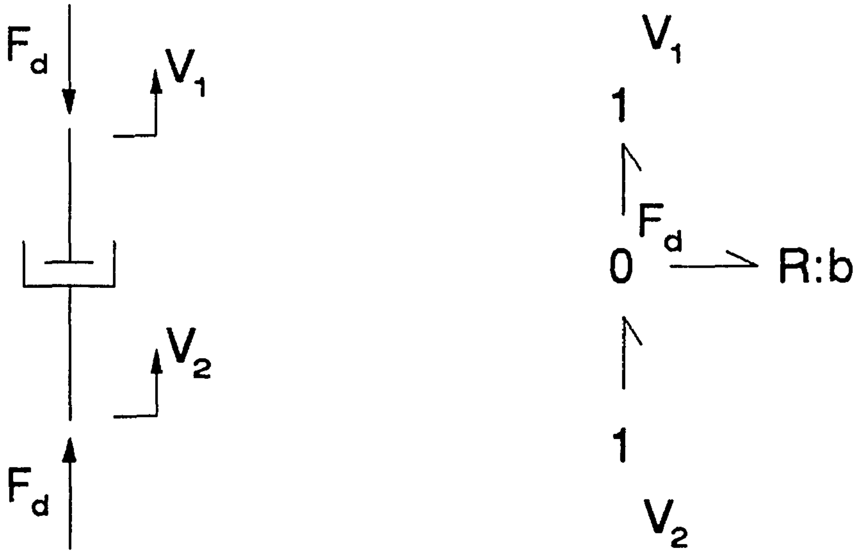


Fig. 2.5. DMA - Linear Translational Damper

SD Spring and Damper **Parameters: B,C,Q_0**

A combination of SPA and DMA; typical application for this component is the modelling of a vehicle passive suspension. Schematically shown in Fig. 2.6, this component's constitutive equation for the force is given by the sum of Eqs. (1) and (2).

FGN Force Generator Parameters: -none-

In order to allow the user to experiment with the widest possible range of suspension schemes, CANVAS provides an arbitrary force generator component. This is a generalization of the two-port force generators such as springs, dampers and cylinders in that the user may substitute any force generation equation desired. Thus, there is no pre-defined component constitutive equation. We demonstrate the concept with an example in Chapter 4.

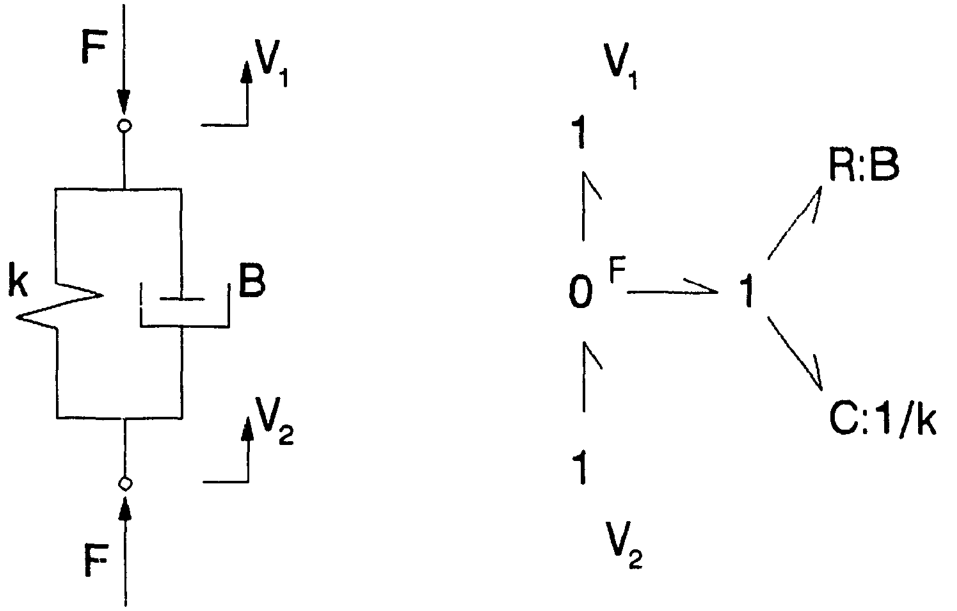


Fig. 2.6. SD - Linear Translational Spring/Damper

2.4.3 Hydraulic System Components

ACC Gas-Charged Accumulator Parameters: V_0, P_0, n

The gas-charged accumulator is modelled as a capacitive element with a polytropic expansion law relating the gas volume and pressure. Referring to Fig. 2.7, the (integral causality) constitutive law reads

$$P = P_0 \left(\frac{V_0}{V} \right)^n$$

where

$$V = V_0 - \int_{t_0}^{t} q \cdot dt$$

and

P	gas pressure
P_0	precharge pressure
V_0	initial volume
n	polytropic expansion constant
q	net flowrate into the accumulator

The bond graph structure for this component is an annotated C-element (C:polytropic), with an attached 1-junction. At the time that equations are written from the bond graph, the annotation is interpreted and the constitutive law substituted in. The presence of the 1-junction ensures that the sign convention is respected.

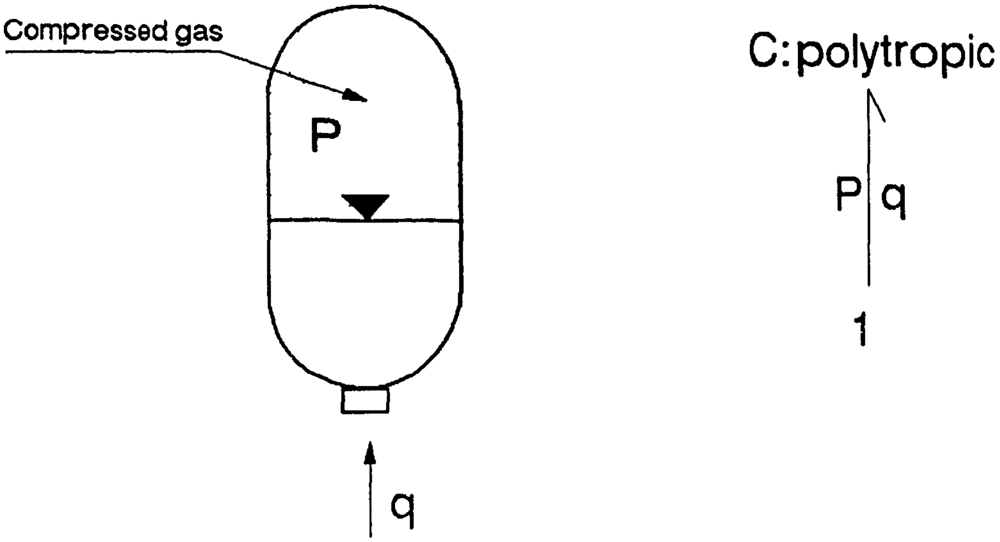


Fig. 2.7.   ACC - Hydro-Gas Accumulator

JUN Hydraulic Junction **Parameters: -none-**

This is a special component, with regards to the pre-processor, as it was explained; connection of a fluid line to another fluid line automatically creates such a component. It is modelled as an ideal junction, and for this reason, the bond graph representation is a simple 0-junction. An instance of its appearance is to be found in example *DOMINY3*, in Chapter 4.

FVL Flow Control Valve **Parameter: R**

The flow control valve (shown in Fig. 2.8) is a dissipative device that restricts fluid flow. Its behaviour varies with the regime of fluid flow (laminar or turbulent). For laminar flow the constitutive equation gives a linear relation between the pressure drop and flowrate:

$$P_1 - P_2 = \frac{q}{R} \qquad \text{(resistance)}$$

$$q = R \cdot (P_1 - P_2) \qquad \text{(conductance)}$$

where the parameter, R combines the effects of viscosity (μ), and conduit length (l) and diameter (d); i.e.

$$R = \frac{128\mu l}{\Pi \cdot d^4}$$

For turbulent flow,

$$P_1 - P_2 = \text{sgn}(q) \cdot \left(\frac{q}{R}\right)^2 \qquad \text{(resistance)}$$

$$q = R \cdot \text{sgn}(P_1 - P_2) \cdot \sqrt{|P_1 - P_2|} \qquad \text{(conductance)}$$

where

$$\operatorname{sgn}(x) = \begin{cases} +1, & x>0 \\ 0, & x=0 \\ -1, & x<0 \end{cases}$$

Both causalities are part of the knowledge-base of the CANVAS system (BOND). When it is time to prepare the equations for simulation, the appropriate form is chosen according to the assigned causality.

The resistive parameter R embodies orifice area (A), discharge coefficient (C_d) and fluid density (ρ), as follows:

$$R = A \cdot C_d \cdot \sqrt{\frac{2}{\rho}}$$

RLV Relief Valve Parameters: P_{max},R

The relief valve is normally closed; when a threshold pressure is reached, the valve opens and (ideally) acts as an orifice restriction to flow. Referring to Fig. 2.9, the constitutive law is:

$$q = \begin{cases} R \cdot \sqrt{P_1 - P_2} & P_1 \geq P_{max},\ P_1 > P_2 \\ 0 & \text{otherwise} \end{cases}$$

where P_{max} is the threshold pressure.

Note that this component has only one valid causality (admittance).

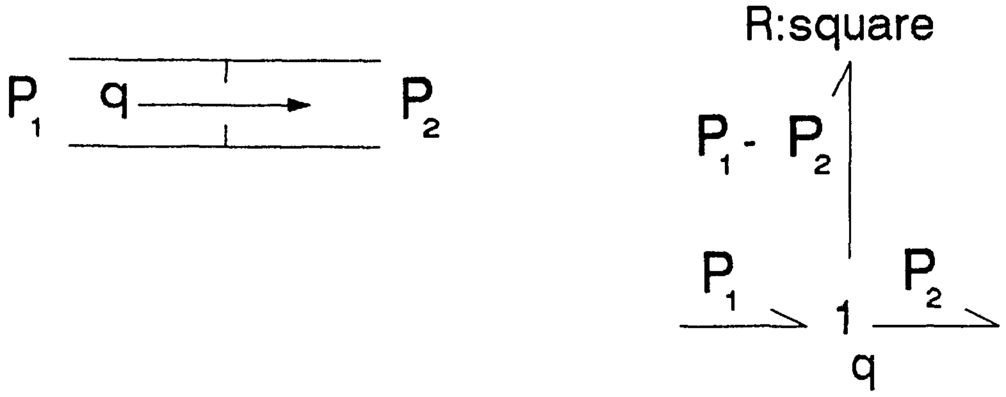


Fig. 2.8. FVL - Flow Restrictor

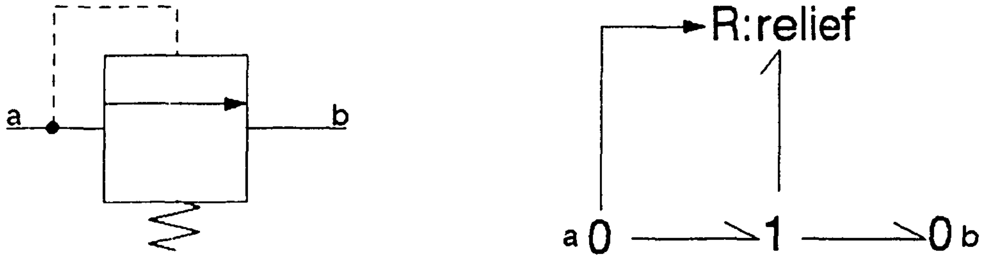


Fig. 2.9. RLV - Relief Valve

CV1 Proportional Control Valve

Parameters: $X_{min}, X_{max}, w, R_{sl}, R_{le}$

The proportional control valve is an essential element in hydraulic control systems. It is used in hydro-mechanical and electro-hydraulic servo systems. It acts as a pair of modulated flow-control valves. The modulation is achieved by the motion of a spool which regulates the size of orifices through which fluid may flow. The schematic and bond graphs for this component appear in Fig. 2.10. The ports (*s*-supply, *e*-exhaust, and *l*-load) are labelled according to the expected hydraulic connections.

For simplicity, the valve is assumed *critically lapped and square-ported*, so that valve area gradient is constant. Different porting characteristics can be implemented at a later stage, when the emphasis of a particular study focuses on control valve influence on the dynamic behaviour.

The full equation for the valve is

$$\left.\begin{aligned} q_{sl} &= R_{sl}\cdot x_v\cdot \mathrm{sgn}(P_s - P_l)\cdot\sqrt{|P_s - P_l|} \\ q_{le} &= 0 \end{aligned}\right\} \quad x_v \geq 0$$

$$\left.\begin{aligned} q_{sl} &= 0 \\ q_{le} &= R_{le}\cdot x_v\cdot \mathrm{sgn}(P_l - P_e)\cdot\sqrt{|P_l - P_e|} \end{aligned}\right\} \quad x_v < 0$$

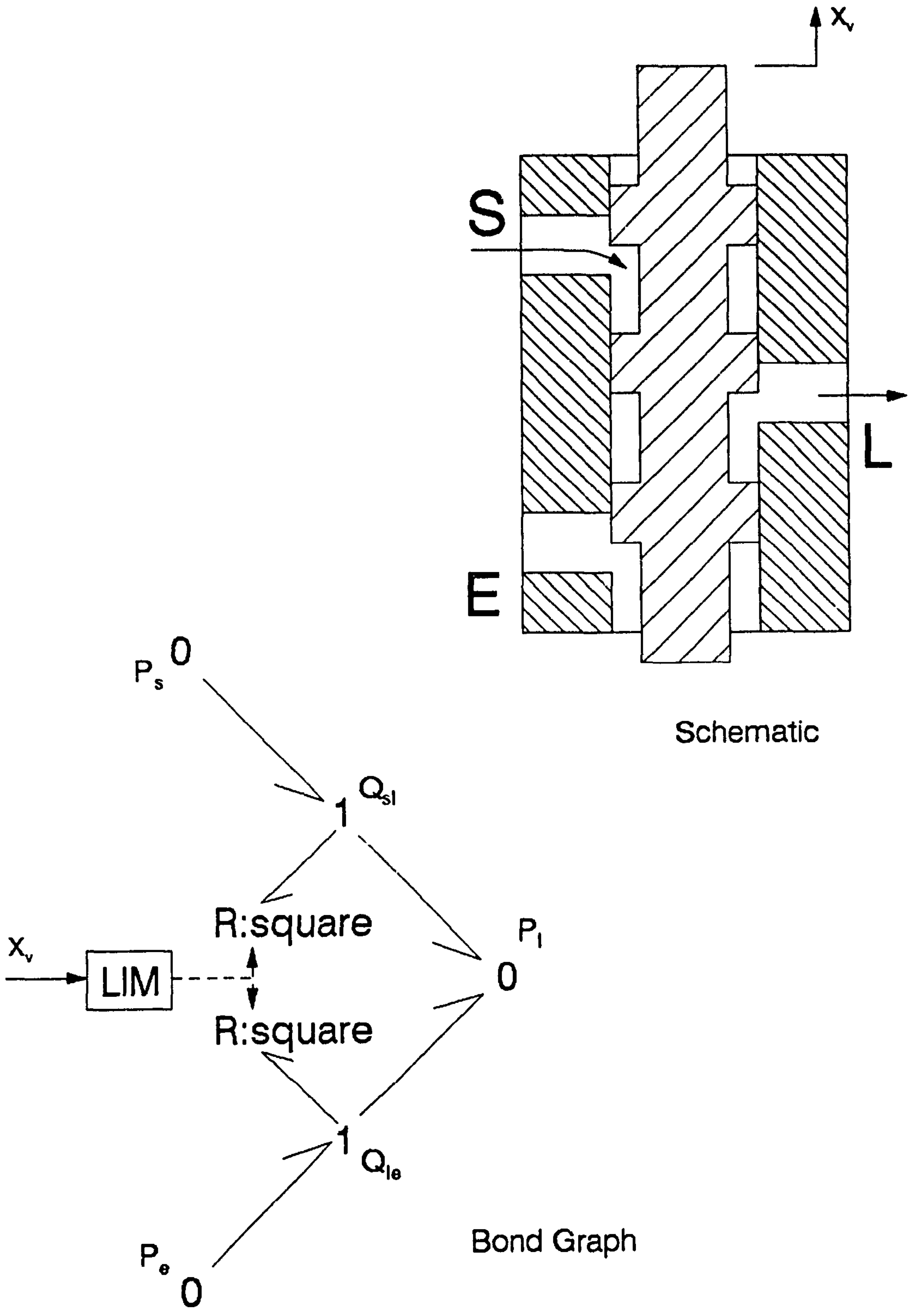
where

$$X_{min} \leq x_v \leq X_{max}$$

The parameters R_{sl} and R_{le} are resistive coefficients as in the FVL component.

The limiter (LIM) block represents the finite valve stroke, which places an upper limit on orifice size, and hence maximum flowrate for a given pressure drop. The parameters X_{min} and X_{max} are the stroke limits on the negative and positive side of null, respectively. Such a limitation on spool travel means that the low gain of the valve becomes zero when the valve is fully opened or closed, and this (saturation) nonlinearity can result in reduced performance in closed-loop control. Because of the potential for non-negligible physical consequences, the limiting effect needs to be included in the valve model.

Note that causality is restricted for this component. It is expected to act as a flow regulator - with pressure being imposed externally on all three ports. Further, note that reverse flow is not prevented, and that if the load pressure should exceed the supply pressure while the valve is stroked positively, then a trouble condition may exist (reverse flow through the control valve); this is avoided by proper system design.



Schematic

Bond Graph

Fig. 2.10. CV1 - Three-Port Control Valve

CY1 Single-Acting Cylinder Parameters: A_h, A_r [$, P_0, V_0, \beta$]

This component, shown in Fig. 2.11 essentially acts as a means to transform energy between hydraulic and mechanical forms. Lossless cylinders can be represented by a simple bond graph TF (transformer) element which embodies the two equations

$$F = A_h \cdot P_h$$

$$q_h = A_h \cdot (V_{ram} - V_{case})$$

Inclusion of physical effects such as fluid compliance and friction losses (conditional expansion) is accomplished by augmenting the basic CY1 bond graph with the appropriate additional structure.

For fluid compliance we require a C:bulk-element attached to the 0-junction representing head pressure. The constitutive law for this compliance effect involves the fluid bulk modulus β and the entrapped fluid volume V. It reads

$$P_h = P_0 + \Delta P$$

where

$$\Delta P = \beta \frac{\Delta V}{V_0}$$

$$V_0 = \text{initial fluid volume}$$

$$\Delta V = \int q \cdot dt = \text{net flow into volume}$$

It should be noted at this point that inclusion of fluid compliance in the cylinder model has some implications for the expected causality of the component attached to the

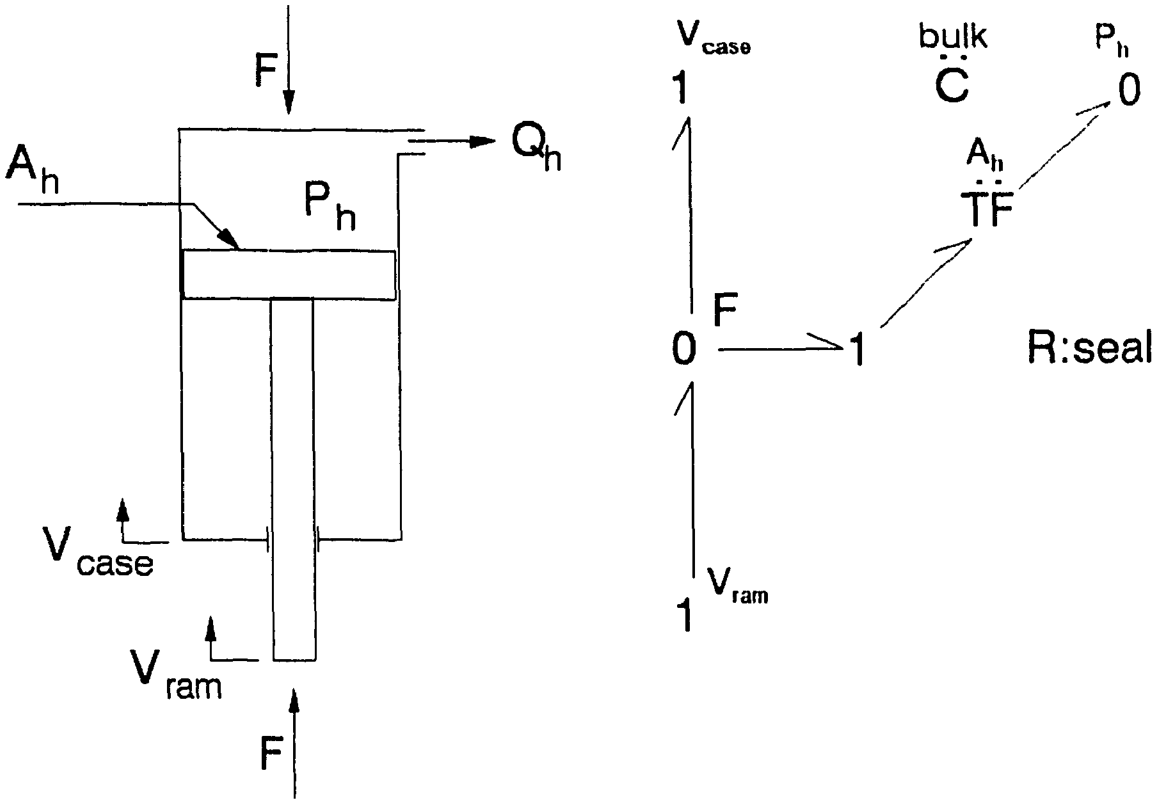


Fig. 2.11. CY1 - Single-Acting Hydraulic Cylinder

head port. Namely, in order to respect integral causality, it is only possible to attach a flow source to the CY1 head port. Conversely, if a flow source is attached to the head port, and if the two mechanical ports are attached to velocity-producing elements, then it is necessary to include the fluid compliance in order to avoid degenerating the cylinder into an algebraic element. The foregoing discussion should point out why it is desirable to be able to include such physical effects into a model.

Losses from seal friction may be accounted for by appending an R:seal-element to the relative-velocity 1-junction. Both of these optional physical effects are illustrated in Fig. 2.11 in dashed lines.

CY2 Double-Acting Cylinder Parameters: A_h, A_r $[, P_0, V_0, \beta]$

This component differs from the CY1 component in that there is an additional fluid port and fluid chamber. It also has some conditionally expanded structure, including fluid compliance, leakage and seal friction. See Fig. 2.12.

CY3 Three-Port Cylinder Parameters: A_h, A_r

This differential area cylinder is used in interconnected suspension systems. The net area, $A_h - A_r$ and area ratio A_h/A_r parameters can be varied to tune the suspension characteristics. The schematic and bond graph structure for this component are shown in Fig. 2.13.

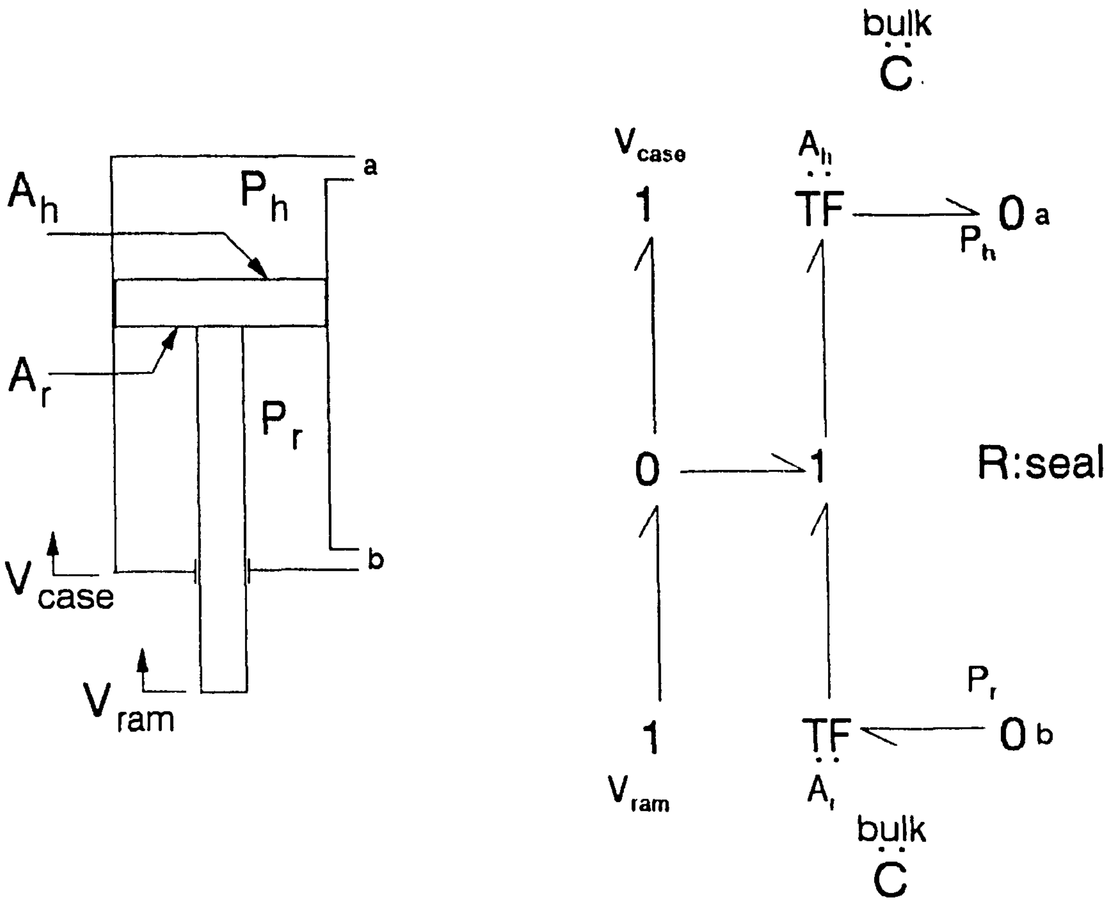


Fig. 2.12. CY2 - Double-Acting Hydraulic Cylinder

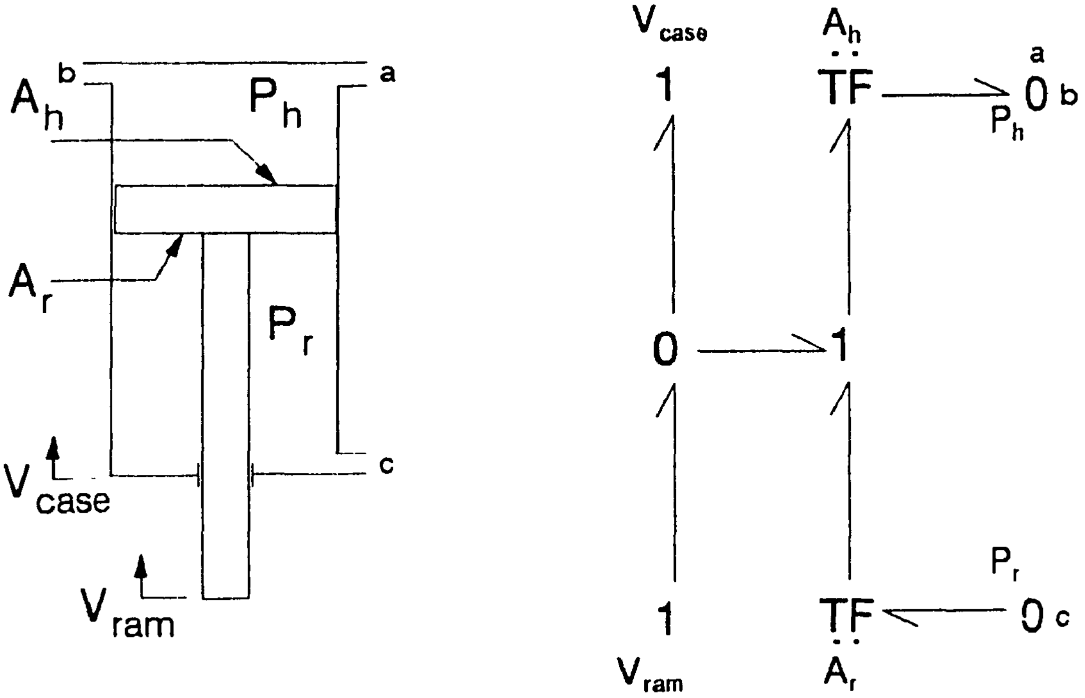


Fig. 2.13. CY3 - Three-Port Hydraulic Cylinder

2.4.4 Transducers

Transducers are an essential part of any feedback control system. For this reason, several transducer elements were included in the package. They are used to interface the mechanical portion of the schematic to the control system portion. The following details existing CANVAS transducer elements, which are shown in Fig. 2.14.

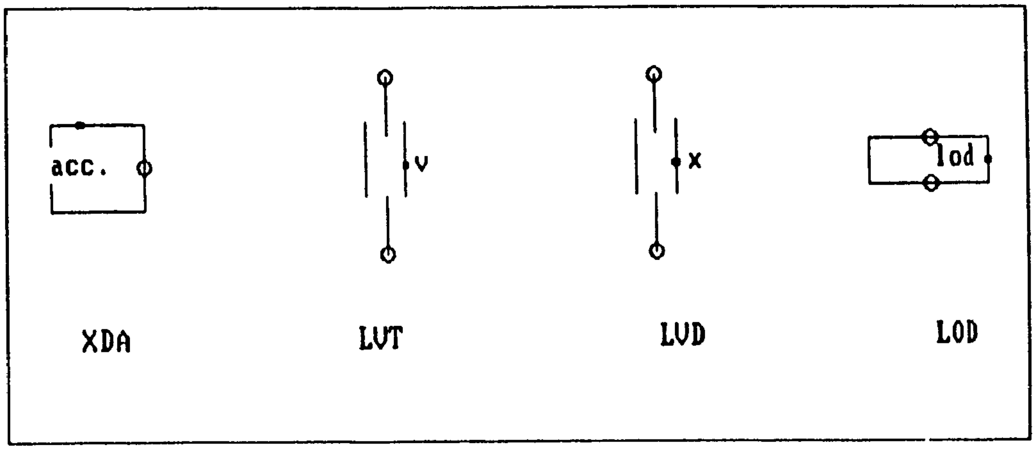


Fig. 2.14. CANVAS Transducers

XDA Accelerometer Parameter: G

The accelerometer, in the current implementation of CANVAS is contrived through bond graph structure operations. It extracts the force signal acting on the 1-DOF mass to which it is attached (and this is the only valid attachment for now), and divides it by the component's mass.

LVT Linear Velocity Transducer Parameter: G

The Velocity Transducer component may be placed between any two valid mechanical ports. Typically, it is placed between a rigid body (M1 or M2) and a reference or other rigid body.

LDT Linear Displacement Transducer Parameter: G

The Displacement Transducer may be used in the same instances as the LVT, with the understanding that its output signal is proportional to the *extension* of the LDT.

LOD Load Cell Parameter: G

Force measurement is achieved by creating a 0-junction with a sink element for the force signal. Load cells may be placed in between mechanical components where a force signal is exchanged.

2.4.5 Block Diagram Elements

CANVAS is a tool for general-purpose model building, and its area of application includes active suspension model development. Thus, the user should be able to construct arbitrarily complex control systems using the same schematic editor used to build physical system models. This is necessary from both a user-convenience point of view and from the standpoint of building a well-integrated and uniform package.

With this requirement in mind, a provision was made within the CANVAS software to support block-oriented control system elements. The user assembles a control system schematic in the same way as for the rest of the system: blocks are placed on the screen and can then be manipulated and connected using simple cursor-controlled commands.

Block diagram elements supported by CANVAS are listed in Appendix 1, along with the remainder of the component models. The CANVAS blocks (and their names) were chosen from a subset of TUTSIM block diagram elements. The reason for doing so is twofold:

- TUTSIM supports a wide range of blocks
- compatibility of CANVAS with TUTSIM is desirable

This compatibility is desirable for verification purposes: a model can be developed using the CANVAS graphical pre-processor, then run using both TUTSIM and CANVAS post-processors. Back-to-back comparison of the time histories should agree closely, except for small differences attributable to the numerical integration scheme.

Block diagram elements can be freely combined with bond graphs according to a few simple rules [22]; the resulting bond graph / block diagram is analyzed in exactly the same way that a bond graph is analyzed. The only difference is that connections in the control system portion carry only one signal, whereas bonds in the bond graph portion carry dual flow-effort signal pairs. A classification of blocks follows, with examples from each class.

Source blocks

These blocks originate signals and are used as inputs to other blocks or to source elements in the bond graph portion of the schematic. They specify given functions of time. A special CANVAS block, XCT is used to give an unspecified function of time which is then chosen by the user at simulation time from a menu of actual source blocks. Example source blocks are

NOI Uniform noise distributed between 0 and 1. This uses available random number generator routines

RPL Rounded pulse with equation

$$RPL(t,\nu,z) = z\cdot\left(\frac{e\nu t}{2}\right)^2\cdot\exp(-\nu t)$$

where

$$z = \text{amplitude},\ e = 2.7182...$$

Memory (storage) blocks

These are blocks that give rise to state variables for the model. Examples are INT, FIO and SEO which correspond to simple integrator, First-Order lag and Second-Order transfer function, respectively. The transfer functions are given as:

$$H_0(s) = s \qquad \text{(INT)}$$

$$H_1(s) = \frac{1}{\tau s + 1} \qquad \text{(FIO)}$$

$$H_2(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad \text{(SEO)}$$

Function blocks

As their name implies, these are blocks that give as output an instantaneous function of the inputs. An example is the LIM (Limiter) block used to model saturation. This block imposes lower and upper limits on the signal entering it. Its defining equation is:

$$y = \begin{cases} X_{min}, & x < X_{min} \\ x \quad , & X_{min} \leq x \leq X_{max} \\ X_{max}, & x > X_{max} \end{cases}$$

An instance of its usage within CANVAS is in simulating control-valve behaviour. The valve's spool travel is essentially limited to a working range dictated by the sizing of the valve by the manufacturer.

Logic Blocks

Logic blocks that perform boolean algebraic functions are included in order to permit the construction of models that embody simple logic (as required, for instance in dissipative semi-active suspensions). As an example of this, consider the diagram in Fig. 2.15, which shows the role of logic blocks in a semi-active suspension.

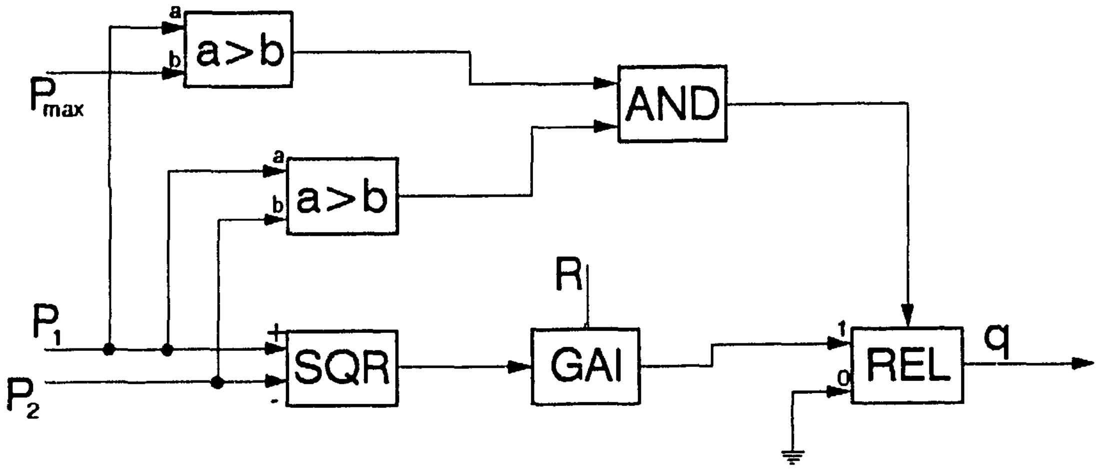


Fig. 2.15. Example Illustrating Logic Blocks

2.5 Summary

This chapter has introduced the modelling philosophy of CANVAS. Modular component models based on bond graphs / block diagrams are the basic units that enable the CANVAS package to model multi-domain systems in a state-variable form.

A sampling of the more important component models was discussed and illustrated; sign conventions and modelling assumptions were stated explicitly.

CHAPTER 3

Software Overview

3.1 Introduction

This chapter discusses the overall software organization of the CANVAS system, which consists of several modules, and then details the operation of each module.

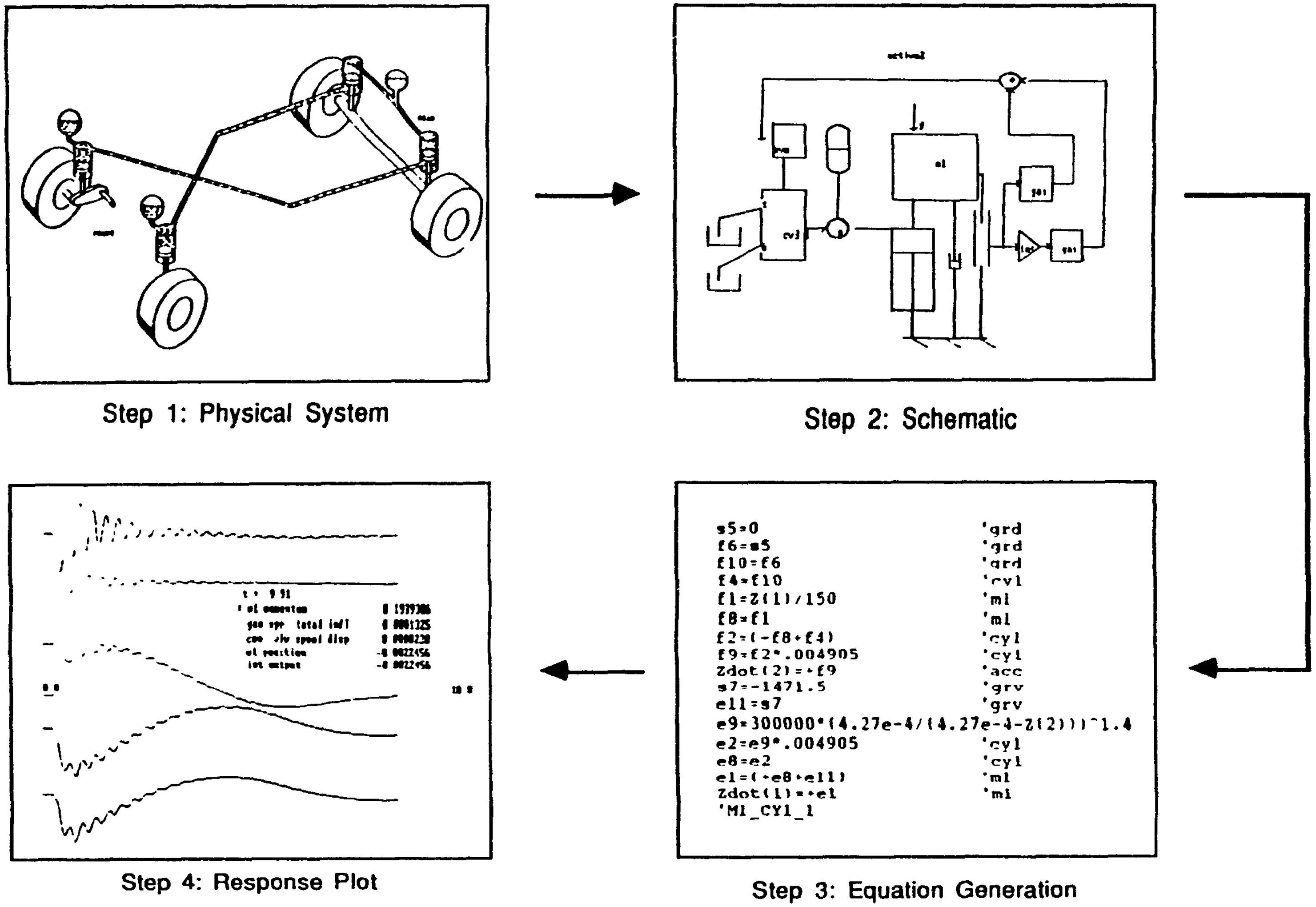
The main purpose in creating the CANVAS software was to enable the engineer to easily build relatively complex system models. For this reason, the user-interface centers around a graphical pre-processor which allows the user of CANVAS to manipulate a model schematic in an intuitive and straightforward way. Building a system schematic requires nothing more than placing component symbols (icons) on the screen, and specifying connections among them.

Using an electronic sketchpad (schematic editor), the engineer adds icons representing rigid bodies, springs, hydraulic struts, and the like. By interconnecting several of these components on the graphics screen of a personal computer, the engineer creates a schematic of the physical system. Component parameter information (e.g. spring compliance) may be entered while drawing the schematic, if so desired. Alternatively, one may build a generic schematic and specify parameter values at a later stage (*deferred* parameters).

Each component in a CANVAS schematic is expanded into a corresponding bond graph / block diagram model. Thanks to the modularity of the bond graph approach, the resulting overall system bond graph is a well-defined mathematical model of the physical system represented by the schematic. It can therefore be readily processed into a set of algebraic/differential equations, which are solved using numerical integration to yield the system's response.

An important aspect of the software package is that the equation-writing process is fully automatic, so that the user of the software does not need to perform error-prone algebraic manipulations. The only requirement is that the user be capable of constructing a schematic diagram of the system so as to meaningfully represent the physical system under study. Numerous checks built into the software aid the user in constructing models.

Fig. 3.1 depicts the overall functioning of CANVAS; it shows a typical physical system (taken from [39]), the pre-processor screen with a schematic of an active suspension system, followed by a set of differential equations, and finally, a response plot. These stages are related to the various CANVAS program modules in the flowchart of Fig. 3.2. It is apparent that the software performs *transformations* on the model, starting with a schematic, going through a bond-graph representation into a symbolic equation form. These stages are explained in more detail in the follcwing sections.



Step 1: Physical System

Step 2: Schematic

Step 3: Equation Generation

```
s5=0                          'grd
f6=s5                         'grd
f10=f6                        'grd
f4=f10                        'cyl
f1=Z(1)/150                   'ml
f8=f1                         'ml
f2=(-f8+f4)                   'cyl
f9=f2*.004905                 'cyl
Zdot(2)=+f9                   'acc
s7=-1471.5                    'grv
e11=s7                        'grv
e9=300000*(4.27e-4/(4.27e-4-Z(2)))^1.4
e2=e9*.004905                 'cyl
e8=e2                         'cyl
e1=(+e8+e11)                  'ml
Zdot(1)=+e1                   'ml
'MI_CY1_1
```

Step 4: Response Plot

Fig. 3.1. Stages in CANVAS Model Development

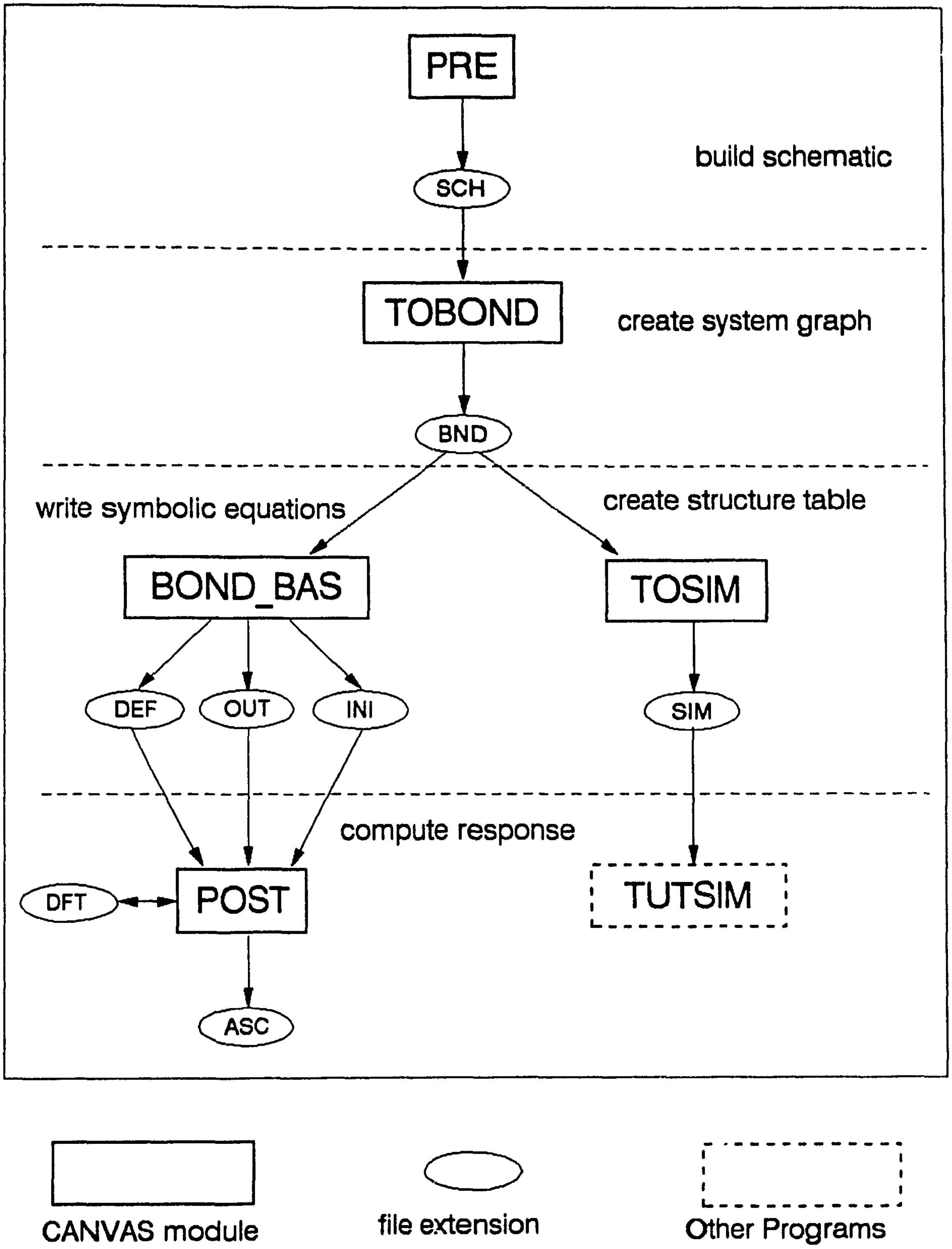


Fig. 3.2. Flowchart of CANVAS Modules

3.2 PRE - Schematic Building

The heart of CANVAS is a graphical pre-processor that allows the user to manipulate various types of icons and thus construct a schematic. When the user is doing this one is able to select pre-defined system components (hydraulic cylinders, control valves, rigid bodies, etc.) from a graphical menu. Fig. 3.3 shows a sampling of mechanical and hydraulic components available within the software. By using appropriate components one may build a great variety of models.

Fig. 3.4 shows several schematics where active suspension components are shown as two-port force generators. Models of this sort are desirable at the conceptual design stage when detailed component descriptions are not available, or are avoided to simplify the model.

In CANVAS, active suspension force-generators can be treated explicitly as separate components (or assemblies). This is important in the context of design, because parameter values are no longer arbitrary entities - they are directly related to some component's physical attributes. Thus, one no longer speaks of gains and time constants, but rather of piston areas and compliances.

Although it has been mentioned that CANVAS does not require the user to know about bond graphs, we have included all of the primitive bond graph symbols in the graphical component library of the pre-processor. Thus, the user familiar with bond graphs benefits from being able to

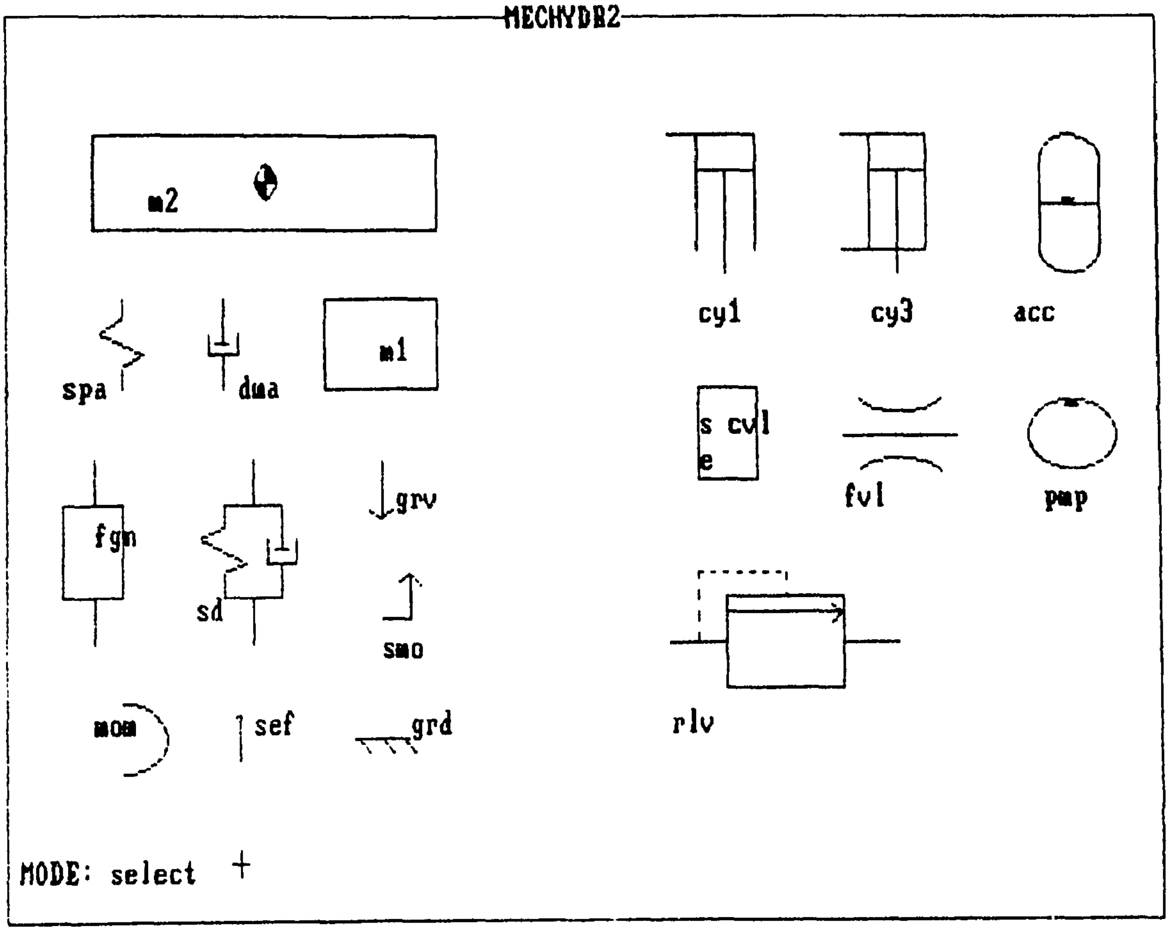
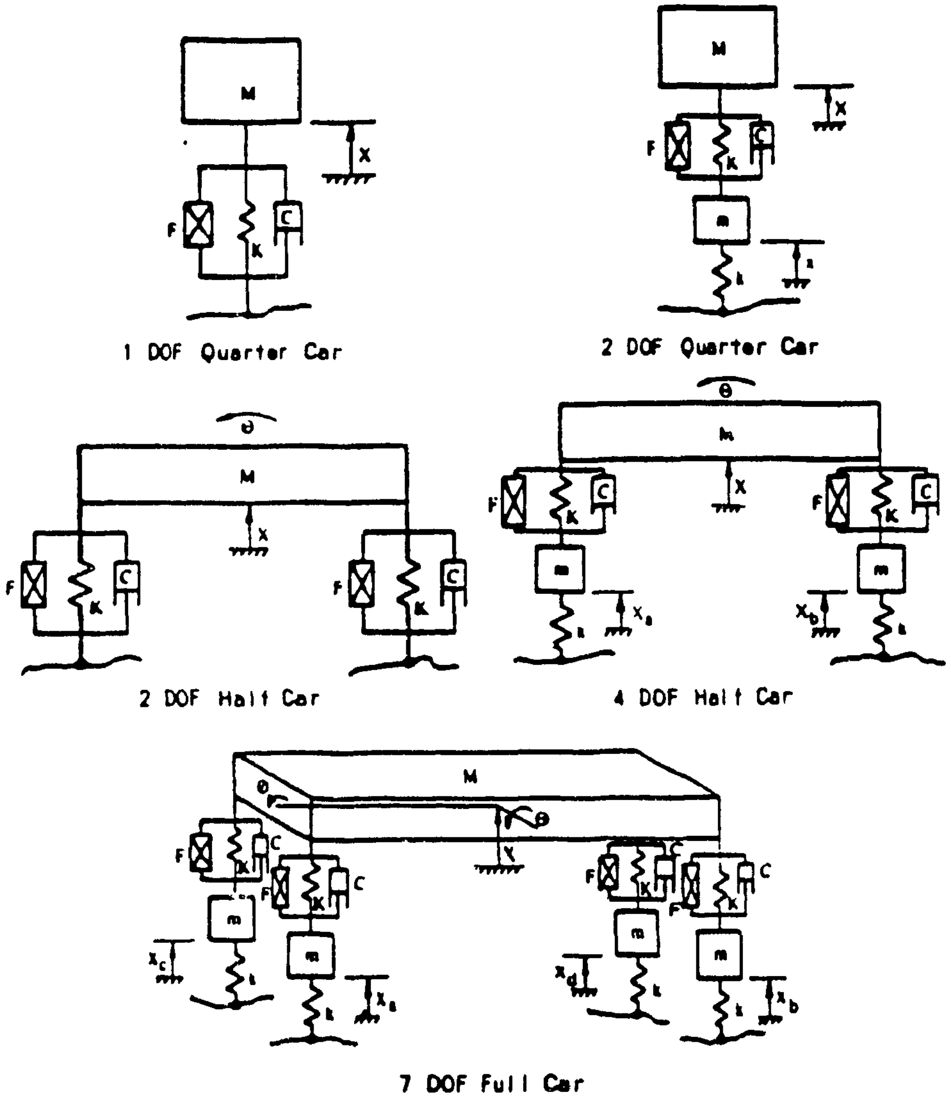


Fig. 3.3. Mechanical and Hydraulic Components in CANVAS



1 DOF Quarter Car

2 DOF Quarter Car

2 DOF Half Car

4 DOF Half Car

7 DOF Full Car

Fig. 3.4. "Black-Box" Representation of Vehicle Suspensions construct arbitrary lumped-parameter models using our package; this is useful when a system must be modelled that comprises components not included in the library of CANVAS.

To facilitate the production of useful schematics, the user is able to modify the screen image in various ways. To this end, graphical transformations were built into the pre-processor to enable the user to manipulate both individual components and the entire screen image.

Component operations

- scale
- translate
- rotate

Screen operations

- zoom
- pan
- border
- toggle port display

Note that CANVAS assumes that these geometric transformations do not affect the resulting system equations. Thus, rotating a rigid body by an angle will not automatically modify the (optional) gravity force acting on the body. This limitation essentially means that it is up to the user of the system to ensure that the schematic makes physical sense, and that the sign conventions assumed by CANVAS (and detailed in Chapter 4) are understood.

Schematic-building is an interactive process in CANVAS. This means that the user can add the desired components to a system schematic and immediately see the results on the graphical screen. The operations available to the user are:

Add
Connect
Delete (not supported at this stage)

When a user adds a component, it is immediately placed on the screen at the current cursor position. It may be dragged (moved) into a new position with cursor movement. It may also be connected with other model components, as described below.

Connections

Within CANVAS, component connections play a central role; this is because the model-building process is essentially a sub-model connection process. While it is possible to have simple models (consisting of uncoupled components), any system of interest will consist of several components interconnected according to a set of rules. The rules are devised in order to maintain model consistency; they are based on an energy-domain port classification. A port *type* is used to characterize the energy domain to which the port belongs. Ports in CANVAS are classified as follows:

MECH	(mechanical)
BOND	(bond)
SIGNA	(signal)
FLUID	(hydraulic)
*	(special)

A connection (between port P_a and port P_b) is valid if P_a and P_b are *compatible*. Compatibility between ports is determined by Table 3.1. If, in attempting a connection the user selects two ports which are incompatible (according to the table), the connection will be rejected by CANVAS.

TABLE 3.1 - Compatibility Conditions

P_b \ P_a	*	BOND	SIGNAL	FLUID	MECH
*	✓	✓	✓	✓	✓
BOND	✓	✓	✓	✓	✓
SIGNAL	✓	✓	✓		
FLUID	✓	✓		✓	
MECH	✓	✓			✓

Assignment of port types is the equivalent of the *regime* assignment that is done in ENPORT. In CANVAS this is done automatically, whereas the ENPORT user must explicitly assign a regime to a bond or node if he or she wishes to distinguish it thus.

As implied in the preceding, it is possible for the user to connect two components at existing (pre-defined) ports. For example, a flow-control valve (FVL) has two pre-defined ports of type FLUID.

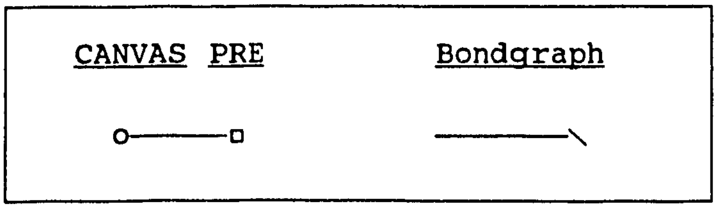
Alternately, the pre-processor creates the necessary port on components where this is required. This is because certain types of components may have an arbitrary number of connection points. An example is the 2 DOF rigid body, M2. Similarly, a single-acting cylinder (CY1) has one pre-defined FLUID port, as well as two attachment points (*head* and *case*) which may be the site of an unlimited number of MECH type ports.

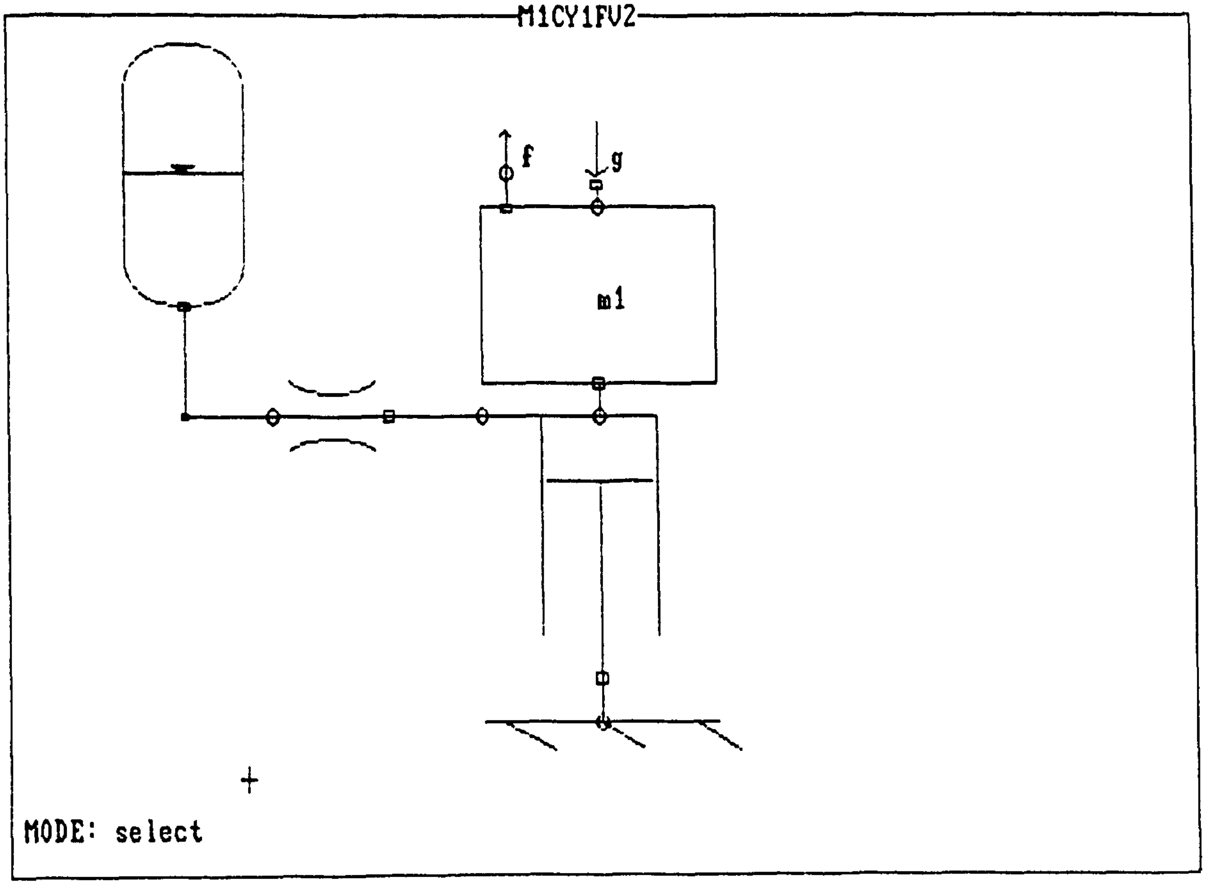
Listed in Table 3.2 by port type are the components on which ports may be created when connecting to another component.

TABLE 3.2 - Components with Arbitrary Number of Ports

Port Type	Component / Element
MECH	m1, m2, cy1, cy2, rb1, rb2, grd, cv3, spa, dma, sd, afg, lod, rvj
SIGNAL	cv2, dif, fio, int, eul, iwz, llg, lmi, lme, rin, seo, spl, abs, fix, att, gai, cos, sin, mul, div, bkl, hys, and, nan, orr, nor, xor, noi, tim, exp, log, lim, max, min, del, pwr, sqt, rsq, gsq, sgn, sum, pid, se, sf
BOND	1, 0, tf, gy, se, r, i, c

Fig. 3.5 shows a CANVAS schematic in which the power ports are displayed (as either a square or a circle). The user of the software can optionally suppress the display of connection ports. This is done in subsequent figures for the sake of clarity. The graphical notation used for power port display within PRE requires elucidation: a connection with a square port at one end and a circular port at the other is the exact equivalent of a power bond, as shown below.





MODE: select

Fig. 3.5. Ports Display in CANVAS PRE-Processor

Finally, the pre-processor has some special procedures to deal with connections. Thus, connecting to an existing fluid segment automatically creates a tee-junction (JUN), and splits the original segment. For the user, this is merely an intuitive operation on a schematic. But it should be noted that the model structure is changed, and the resulting equations are consequently also modified.

3.3 TOBOND - Conversion of Schematic to Bond Graph

3.3.1 System Bond Graph Creation

In order to write dynamical equations for the system whose schematic has been prepared using the pre-processor, CANVAS first converts the schematic into a system bond graph.

The module that translates a schematic into the system bond graph (called TOBOND) is written in PROLOG. The first stage of its operation is to go through the schematic, component-by-component, and to expand the component description into the corresponding bond graph / block diagram structure.

The process of system graph creation is relatively straightforward for simple components, but gets more involved for components with variable parameters. For instance, in the two 2 DOF rigid body, each connection point yields some 'junction structure'.

3.3.2 Representation of System Graph

The CANVAS system graph created by TOBOND consists of PROLOG 'clauses' that declare the existence of *NODES* and *BONDS*. The form of these clauses is given by the prototypes

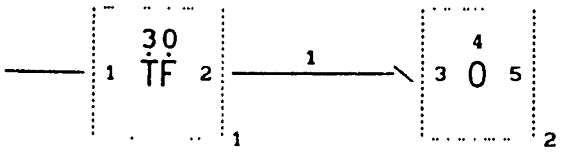
```
node(NodeID,Type,PortList,ParameterList)
bond(BondID,Port,Port)
```

For example, the following clauses

```
node(1,TF,[1,2],[30])
node(2,0,[3,4,5],[])
```

```
bond(1,2,3)
```

represent the bond graph structure in the Fig. below



3.3.3 Stage I - Component Expansion

Simple components (e.g. block diagram elements such as gains and function blocks) are expanded directly. More complex components, such as hydraulic cylinders with multiple models, or the aformentioned 2 DOF rigid body, have special routines for expansion; these were detailed in Chapter 2.

3.3.4 Stage II - Component Connection

The second stage of TOBOND is to perform the component connections, which is straightforward in that connections are treated as simple bonds (for power components) or signal connections (for block diagram elements).

3.3.5 Bond Graph Simplification

Certain bond graph patterns which arise from component interconnections can be eliminated without altering the implied equations. Such is the case with junction nodes having a 'through' sign convention [2]. The module TOBOND, after construction of the system graph, invokes a simplification procedure which removes redundant bond graph structure. This simplification simply improves the efficiency of the resulting numerical simulation by eliminating redundant equations that must otherwise be evaluated at each integration step.

As an example, consider the sequence shown in Fig. 3.6. The structure (a) results from the assembly of an accumulator with a flow-control valve; it contains two redundant 'through' nodes. These are eliminated by the simplify procedure, which detects such a pattern. The result, shown in (c), requires fewer equations to be written.

It should be recognized that this symbolic manipulation of the bond graph is entirely equivalent to the symbolic manipulation of the resulting equations, which is performed by such programs as MACSYMA. It is not intended here to imply that the power of our approach is even close to that of a true symbolic math package; however, it is apparent that manipulations of the bond graph structure by the computer are possible and useful.

3.3.6 Causality Assignment

Although it figures so prominently within the bond graph framework, "causality is not a natural fact, but selected by the engineer to facilitate computation" [38]. Nonetheless, causality assignment of the system bond graph is an essential step toward the generation of dynamical equations [8]. The causally augmented bond graph is an unambiguous representation of the system dynamical equations; indeed it is an 'algorithm' [8] to compute the variables of the system. In [9], a causally augmented bond

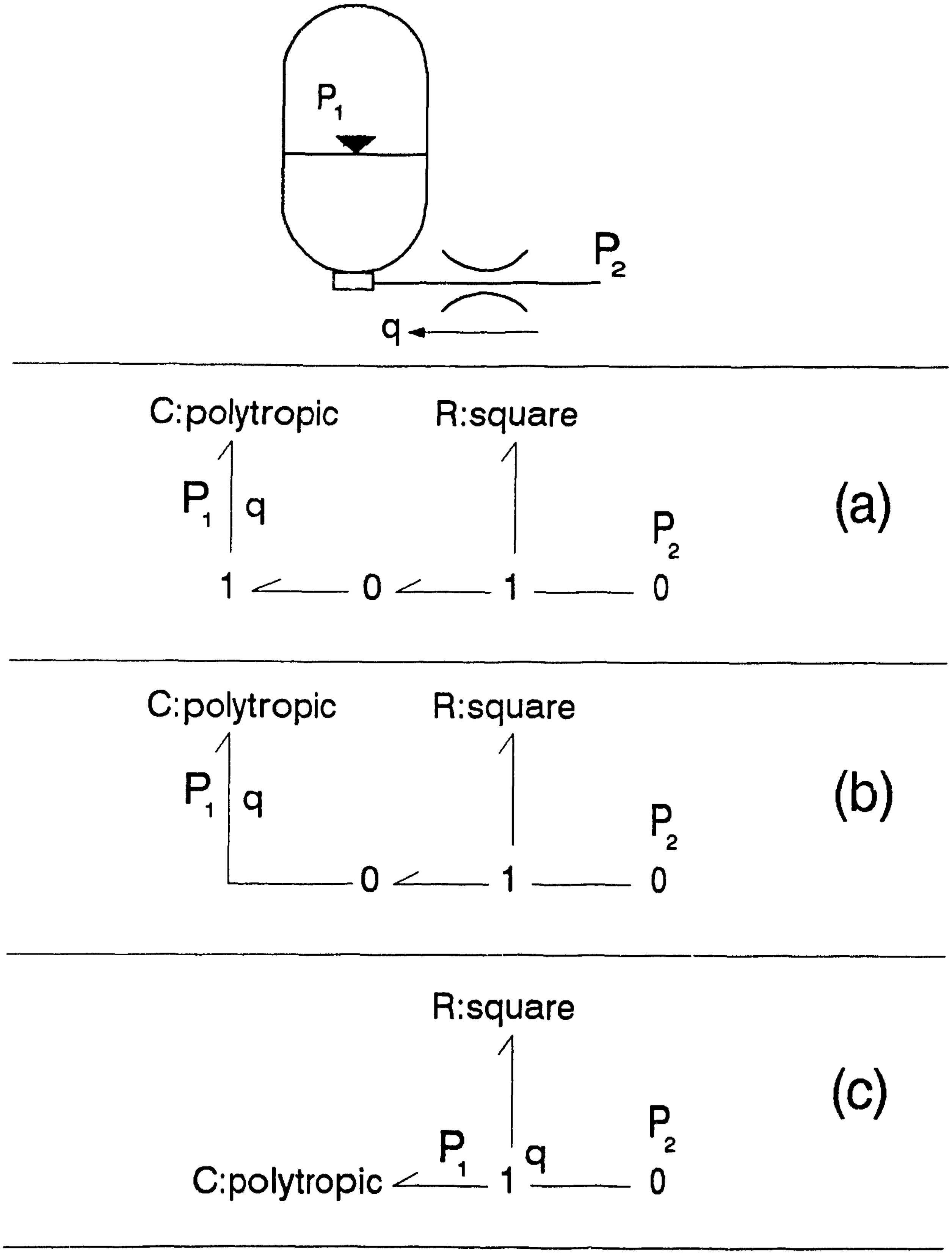


Fig. 3.6. "SIMPLIFY" Procedure Applied to Sample Bond Graph

graph is said to be equivalent to a computing diagram.

Simply put, causality can be thought of as the direction of an assignment statement in a sequential execution. For instance, in FORTRAN, the statement

A = B (as opposed to B = A)

means that A is causally determined by B.

The procedure used for causality assignment within CANVAS is the SCAP (Sequential Causality Assignment Procedure) discussed by Rosenberg and Karnopp [2]. It is a constrained propagation of causality within the bond graph. Causality constraints arise from different considerations:

- by necessity (or definition)

SF	flow-out
SE	effort-out
0	single effort-in
1	single flow-in
TF	one flow-out; one effort-out
GY	two flow-out OR two effort-out

- for computational convenience

C	effort-out
I	flow-out
R:check	flow-out

These two classes of causal constraints need elucidation. In the first case we are dealing with hard constraints that arise from the definition of the nodes themselves. For instance, it makes no sense to introduce a flow source (SF) and then attempt to impose a flow on it.

In the second class of constraints are those that arise from a particular choice of element constitutive equation, which is preferable for numerical computation. Thus, as it is preferable to numerically integrate rather than differentiate, storage nodes have a preferred causality - the integral. In the same vein, an element such as a check-valve with no dynamics is conveniently treated in one causality, but not in the other.

Hood et al. [40] describe an implementation of the SCAP algorithm based on a state-transition table. A similar approach is used within TOBOND.

The outcome of the SCAP is a fully augmented bond graph which, when in full integral causality, describes a unique set of explicit differential/algebraic equations. This type of equation set can be numerically integrated in a straightforward manner.

Failure of the SCAP to produce full integral causality can result from:

- Implicit R-Fields
- Dependent I-elements (e.g. constrained inertias)
- Dependent C-elements (e.g. compliances in series)

When causality assignment fails to produce a fully explicit set of state equations, there exist techniques, both numerical and analytical that can be used to solve the set of implicit equations [13]. Hood et al. [9] describe an algorithm to treat the calculation of implicit R-Fields. We ignore such refinement for the present, and if the SCAP fails, the user is warned, at which point the model should be changed before proceeding with the analysis.

3.4 BOND_BAS - Automatic Equation Generation

The bond graph describing the overall mathematical model of the system is unambiguous when it is augmented with causality information.

The equation generation module (BOND_BAS) assumes that the system bond graph can be assigned full integral causality by the TOBCND module. The equation-writing procedure is based on that given by Margolis [41], and is also analogous to that used by Granda [23]. This procedure consists of traversing the bond graph according to causality, starting with each storage node. As each node is reached, its constitutive equation is written. These component equations were detailed in Chapter 2.

The reason why such an algorithm works is that a bond graph in full integral causality, is equivalent to a Directed Acyclic Graph (DAG). Each state variable's derivative can be expressed as an *explicit* function of inputs and other state variables. In mathematical notation:

$$\dot{x} = \Phi(x,u)$$

The presence of algbraic loops is equivalent to an implicit equation of the form

$$\dot{x} = \Phi'(x,\dot{x},u)$$

An example of a DAG is shown in Fig. 3.7, along with the corresponding equations. As is apparent, CANVAS writes equations in first-order state-variable form.

```
f5 = Z(1) / 150             'm
Zdot(2) = f5                'spring
e4 = Z(2) / .0000027        'spring
e9 = (-e4)                  'spring
e8 = xct(1, t)              'fext
e1 = (e8 + e9)              'm
Zdot(1) = e1                'm
'M1
```

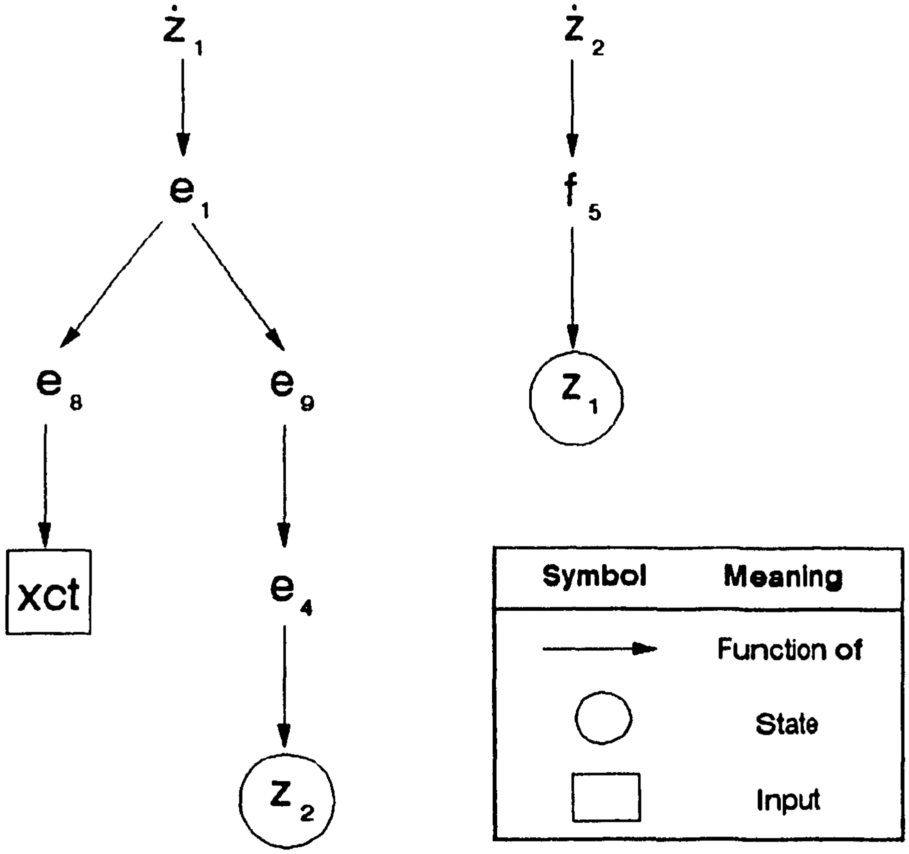


Fig. 3.7. Directed Acyclic Graph

Self-Documentation

The usefulness of automated equation-writing can be enhanced if such equations are written in a symbolic form easily interpreted by humans, rather than being in a cryptic form (as in the program NEWEUL [42]). In CANVAS, ease of interpretation is achived through

- writing the equations symbolically, in BASIC
- automatically adding comments, to indicate the comonent from which each equation originates
- creating additional BASIC program statements expressing the meaning of state and output variables (state-table).

The benefits of this are

- manual verification of equations is possible
- equations may be manually modified (see example THOMP155, p. 138)
- the post-processor can automatically label the time histories

3.5 TOSIM - Interface to TUTSIM

An alternative to writing symbolic equations in a language such as BASIC is to use TUTSIM [20,21] to solve the CANVAS model. This alternative is provided by an interface module, TOSIM, which creates a structure table compatible with the TUTSIM package from the fully augmented bond graph.

The ability to use TUTSIM as a solution stage for CANVAS has some advantages; chief among these is the ability to verify the numerical code, and the possibility of comparing performance of the two packages. Another way of looking at it is that the CANVAS pre-processor may be considered as a pre-processor for TUTSIM as well.

3.6 POST - Numerical Integration Post-Processor

CANVAS's solution stage is an interactive graphics program used for performing the numerical integration and response plotting of the previously developed models. It differs from conventional numerical integration software in that the response plot may be interactively altered during the computation.

Thus, the user may rescale or shift the plot vertically or horizontally, and may decide to continue the simulation after it has been interrupted. Typically these features are useful for debugging a system simulation for the first time, when its behaviour is unknown, and a-priori decisions on duration of the simulation time and relative scaling of the plotting variables cannot be reliably made.

Features and Restrictions:

Numerical Integration (fixed stepsize RK4)
Interactive display control
Frequency Analysis (Fast Fourier Transform)
Context saving and loading
Output to ASCII file
Parametric Study (Multi-Run)

Numerical integration of dynamical equations is the central purpose of the post-processor. Thus its performance is intimately tied to the numerical code used. The choice of a fixed-step Runge-Kutta of order four (RK4) integrator was motivated by two factors: simplicity and speed.

Fixed step size integrators have some inherent disadvantages. For instance, numerically stiff systems

(systems with widely spaced eigenvalues) necessitate small step size to track fast-changing quantities, and long runs to characterize the behaviour of slow-moving signals. Thus simulation becomes computationally intensive for these systems. There exist algorithms that employ variable step size integrators expressly designed for this problem, (Gear [43]) but we leave implementation of such algorithms to future enhancements of the software.

Interactive Display Control

The capability to produce graphical output from numerical integration is commonly available with practically any package. What is generally missing from packages is a set of functions designed to enhance the modeller's speed at *interactively* producing the *desired* output. Such a set of features was designed and implemented in CANVAS's post-processor; it enables the user to quickly and effortlessly modify a plot even as it is being produced. The user has control over several aspects of the response plot:

- display list
- scaling of each variable (automatic and manual)
- alphanumeric labels
- vertical and horizontal rulers
- legend
- line style and color

The implementation of the post-processor is such that the user may perform any of the functions at any time, through the use of pop-up menus. A built-in help menu facilitates the task for the user unfamiliar with the program.

Fast Fourier Transform (FFT)

Time-response plots are adequate for evaluating certain aspects of a physical system's dynamic behaviour. In vibration studies, as well as in other contexts, a frequency analysis of the time response often provides valuable additional insights.

For instance, it is well known that vibratory systems under free vibration will exhibit behaviour combining the natural modes of the system. It is then desirable to have plots of frequency-dependent response for physical systems, for both software verification and the analysis of models.

Frequency domain analysis is also one of a few ways of obtaining the stochastic response of nonlinear systems. Monte-Carlo simulation using random inputs, such as synthetic white noise, is used. The response of the nonlinear system is then FFT analyzed to identify dominant modes.

To endow CANVAS with the ability to perform frequency analysis, a Fast Fourier Transform algorithm was implemented in the post-processor. The FFT is a well-known method of evaluating the (discrete) frequency content of a periodic time history. Since the system response is evaluated as a discrete-time state trajectory, the FFT can provide a frequency-domain representation of the system response.

Deferred Parameter Entry

The specification of model parameters is often the result of a design process done by the engineer who is seeking acceptable system performance. While some system parameters may be known a-priori (fixed), some may be arbitrary, or the subject of a study. In the spirit of providing a user-friendly package, CANVAS's parameter-entry was designed with this in mind. The user of the system may enter parameters in either of these stages:

- PRE-processor

the parameter becomes associated with the model schematic, as well as all subsequent steps. Alteration of the parameter value requires manual editing of the equation file (*.OUT) for the model. However, doing this is not recommended since it introduces inconsistencies between the system bond graph and the equations.

- POST-Processor

deferral of parameter specification requires the user to input parameter values at each simulation, which is useful for conducting trial-and-error runs. They may be saved in a defaults file. Alternatively, the Multi-Run capability may be used, as discussed below.

Multi-Run Capability for Parametric Studies

When studying complex, non-linear systems, an often used method of obtaining meaningful results is to run several simulations while varying one system parameter. One or more time-histories are then evaluated against this varying parameter, and conclusions may be drawn. This multi-run capability exists in other packages (e.g. TUTSIM) and it was deemed desirable in CANVAS since intended applications of the package include parametric studies.

To obtain a multi-run for a particular parameter, the user selects the desired parameter from the list of deferred parameters, and enters its initial and final values and increment. The user also selects a plotting variable from which a particular value (e.g. maximum, or mean) is to be extracted.

The post-processor then proceeds by repeating the simulation for each value of the multi-run parameter, while extracting the values the user is interested in. After the multi-run is completed, a plot of the extracted value versus the parameter value is displayed.

Other Features

The POST-PROCESSOR has some other features useful in producing graphical output.

- Indicator lines

  the user may add indicator lines to show levels of interest in a particular trace.

- Computation of mean, minimum and maximum

  the user may invoke these features to compute and display a quantity of interest for a given trace. An indicator line is added.

Restrictions

Due to memory limitations in the PC, restrictions are placed on the dimension of the state vector's time history.

- Max. size of state vector = 12
- Max. length of time history = 1024 points

By using a file-based storage scheme, the time history could be of arbitrary length, although at the expense of simulation speed.

3.7 Summary

This chapter has presented the software organization of CANVAS. The various modules (PRE, TOBOND, TOSIM, BOND_BAS and POST) were explained. Issues such as component expansion, causality assignment and automatic equation generation were addressed. Software features and restrictions were listed, as they apply to each module.

CHAPTER 4

Software Validation and Application

4.1 INTRODUCTION

Thorough testing of a complex software system such as CANVAS is a task which cannot be carried out to the point of ruling out any possible flaws. It is impossible to test the infinite number of combinations of components; thus it is realistic only to verify the operation of the system on several test cases and thereby build confidence in the validity of the software.

Software validation was also an issue in Kinoglu et al. [28] for their fluid power simulation package. In their approach:

> A number of sample circuits have been created to test and validate the performance of each program, by means of both manual calculations and, where feasible, by *comparison with the results of other general-purpose simulation programs.* [italics: present study]

The validation procedure for CANVAS follows this approach, while reflecting the organization of the software. CANVAS models undergo a sequential transformation which was summarized in Fig. 3.2.

It is necessary to verify that each of the transformations is correct in order to ensure that the overall result is correct. The validation is therefore broken down into three phases.

I assembly of components into a system schematic, yields the expected bond graph structure.

II validity of the equations written based on the (correct) structure.

III accuracy of the computed solutions by cross-checking of simulation results with known solutions.

PHASE I

Checking that the expected bond graph structure is created by TOBOND is accomplished by manually constructing a graph from the bond graph structure information produced by TOBOND. This structure, as explained in Chapter 3, consists of PROLOG clauses that declare the existence and connectivity of nodes with bonds and ports.

PHASE II

Because CANVAS utilizes bond graphs as a representation for dynamical equations, it is of interest to verify that the equations derived from system bond graphs correspond to the equations represented by the schematic. The verification of dynamical equations consists of comparison of the equation file produced by BOND_BAS with a set of manually derived equations for the corresponding schematic.

PHASE III

Obtaining confidence in the accuracy of numerical results is a matter of computing the response of systems whose behaviour is known - either analytically, or from the published literature.

4.2 SAMPLE MODELS

In this chapter a number of models are developed using the CANVAS software and simulations are carried out using these models. The principal area of application is hydraulic vehicle suspensions. Certain examples parallel research work published in the vehicle dynamics literature in recent years.

Each of the examples below was selected in order to illustrate some particular feature or component model in the CANVAS system, while simulataneously being useful from a validation point of view. Examples are presented in sections:

OBJECTIVE: the goals of the example are outlined.

REMARKS: textual description of the model(s) used, with reference to schematics.

MODEL: table of components, with a description of the represented physical entities and parameter values.

ANALYSIS: material used to illustrate how

- parameter values are obtained from design considerations
- simulation results are predicted

INPUTS: description of the input functions and/or non-zero initial conditions used to drive the system.

RESULTS: a discussion of simulation results, and correlation with published literature, if applicable.

4.2.1 Radar Pedestal Positioning Servo (ENP13172)

OBJECTIVE

This example is chosen to illustrate the manner in which CANVAS is used by the engineer familiar with bond graphs to create physical system models for arbitrary lumped-parameter systems.

REMARKS

We have duplicated a model given in [22]. The hand-drawn schematic in Fig. 4.1 shows a radar pedestal positioning unit. The ENPORT bond graph is shown in Fig. 4.2. The model includes actuator dynamics, load dynamics, gear reduction, and feedback control. We have duplicated the given bond graph with the CANVAS pre-processor, and this is shown in Fig. 4.3.

MODEL: ENP13172

Component	Parameters	Units	Represents
SEe	1.00	–	Electrical Dynamics
Ie	0.10	H	
Re	5.00	Ω	
1e	-none-	–	
Im	0.25	$kg\cdot m^2$	Motor dynamics
Rm	0.33	N·m·s/rad	
Sem	20.00	–	
1m	-none-		
Cs	0.01	rad/N·m	Shaft Compliance
0	-none-		
TF	30.00	–	Gearing
Ip	320.00	$kg\cdot m^2$	Pedestal
Rp	10.67	N·m·s/rad	
1p	-none-		

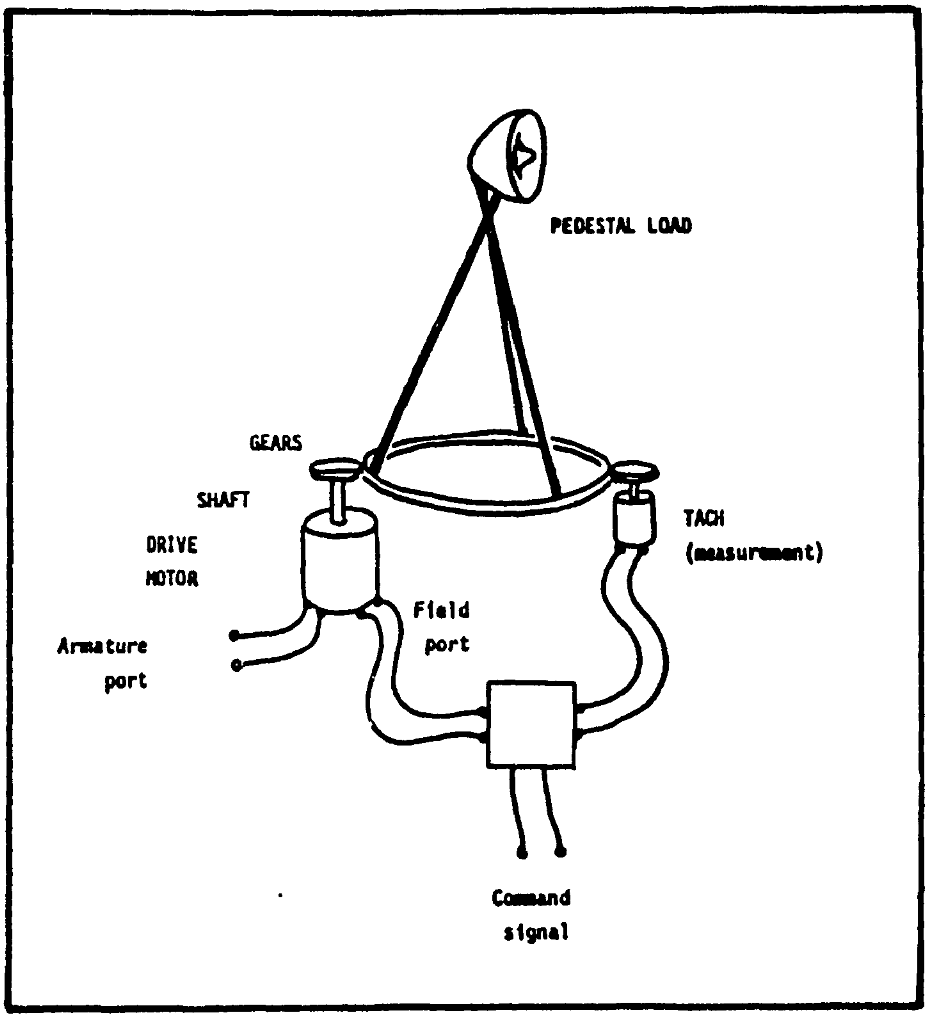


Fig. 4.1. Schematic of Radar Pedestal Positioning Unit

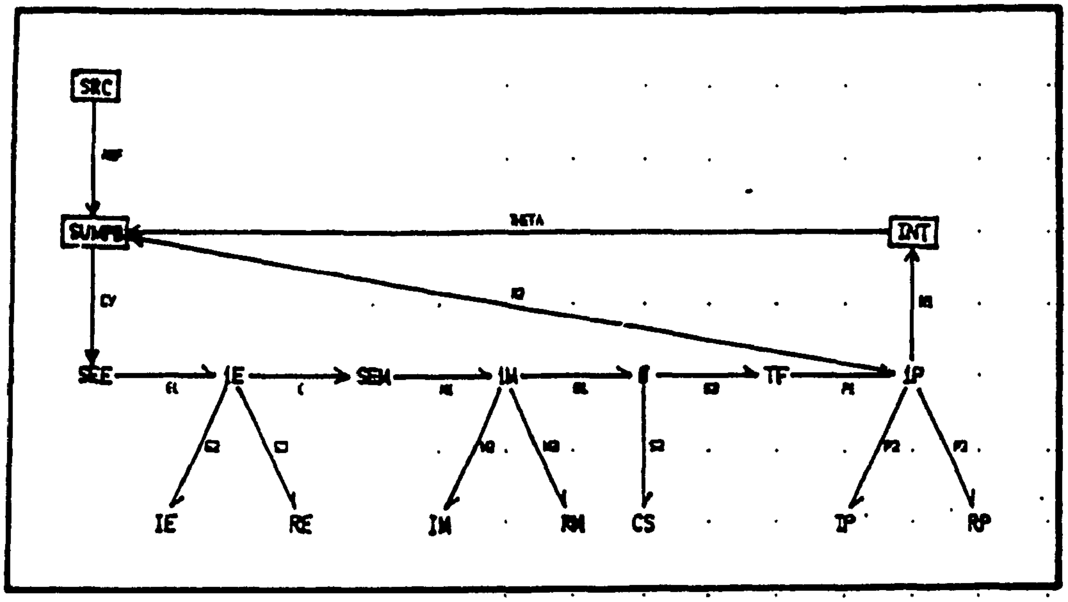
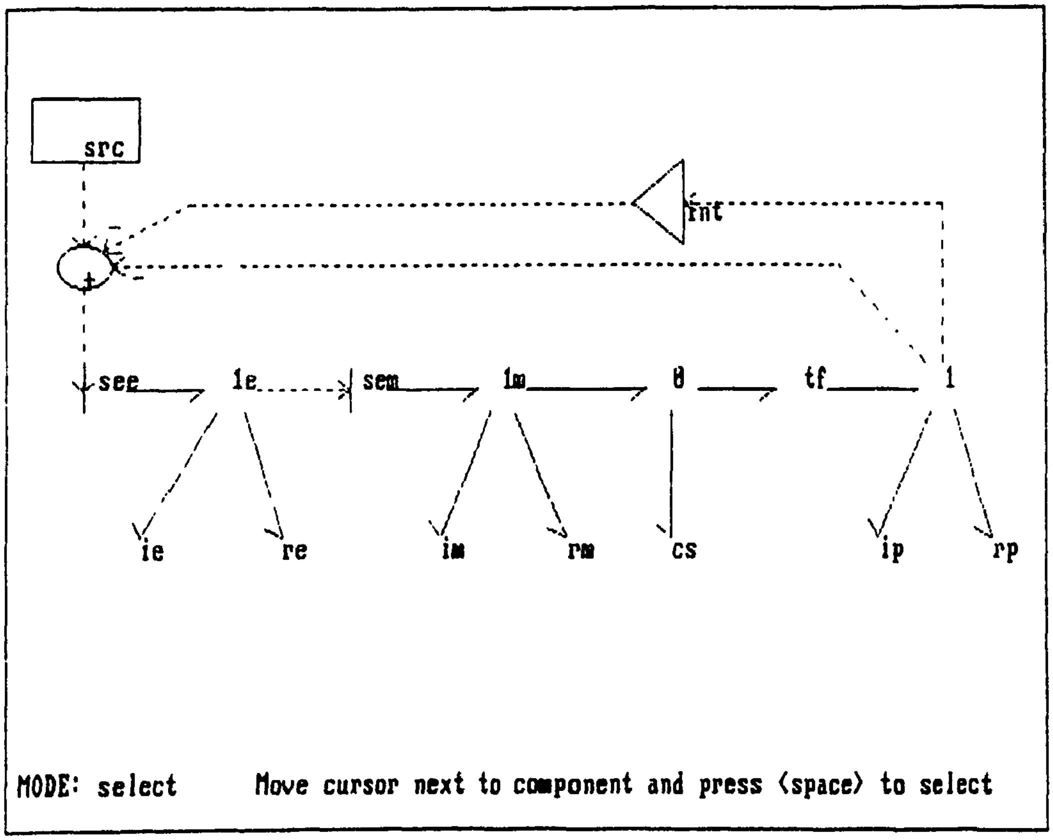


Fig. 4.2. ENPORT-7 Bond Graph for Radar Pedestal Unit



MODE: select Move cursor next to component and press <space> to select

Fig. 4.3. CANVAS Bond Graph for Radar Pedestal Unit

INPUTS

As in the original example, the input is a unit pulse reference displacement; this is provided by the XCT block, which has been relabelled 'SRC' in Fig. 4.3 to correspond with ENPORT's nomenclature.

RESULTS

The response of the radar pedestal unit, as computed using ENPORT-7 is shown in Fig. 4.4. This is to be compared with the response as computed by CANVAS, shown in Fig. 4.5. This latter figure was produced by directly copying the Post-Processor screen to the printer; subsequent CANVAS output has been further processed.

Note that the CANVAS Post-Processor outputs angular momentum (labelled JP*W2 in the plot), as opposed to angular velocity; this is simply a result of the choice of momentum as the state variable for inertia elements.

The similarity in the schematic input and the close agreement in the computed time responses of ENPORT-7 and CANVAS show that our package can easily duplicate the functionality of ENPORT in modelling with primitive bond graph elements. In subsequent examples we will illustrate how the primitive bond graph description is bypassed, thus demonstrating the real benefits of the CANVAS approach.

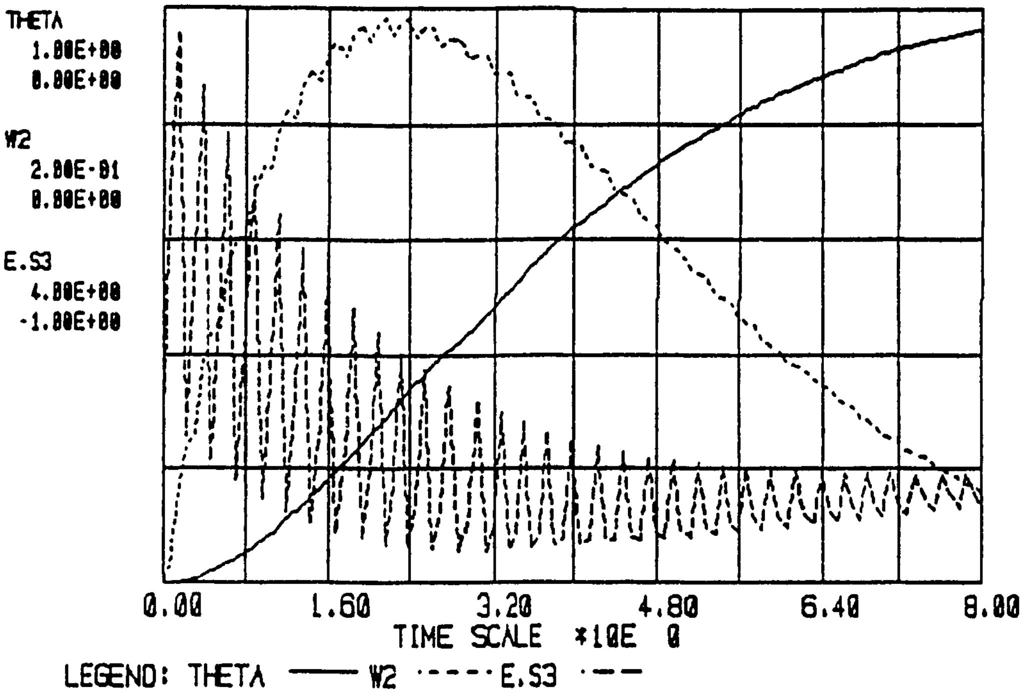


Fig. 4.4. Response of Radar Pedestal Unit Computed by ENPORT

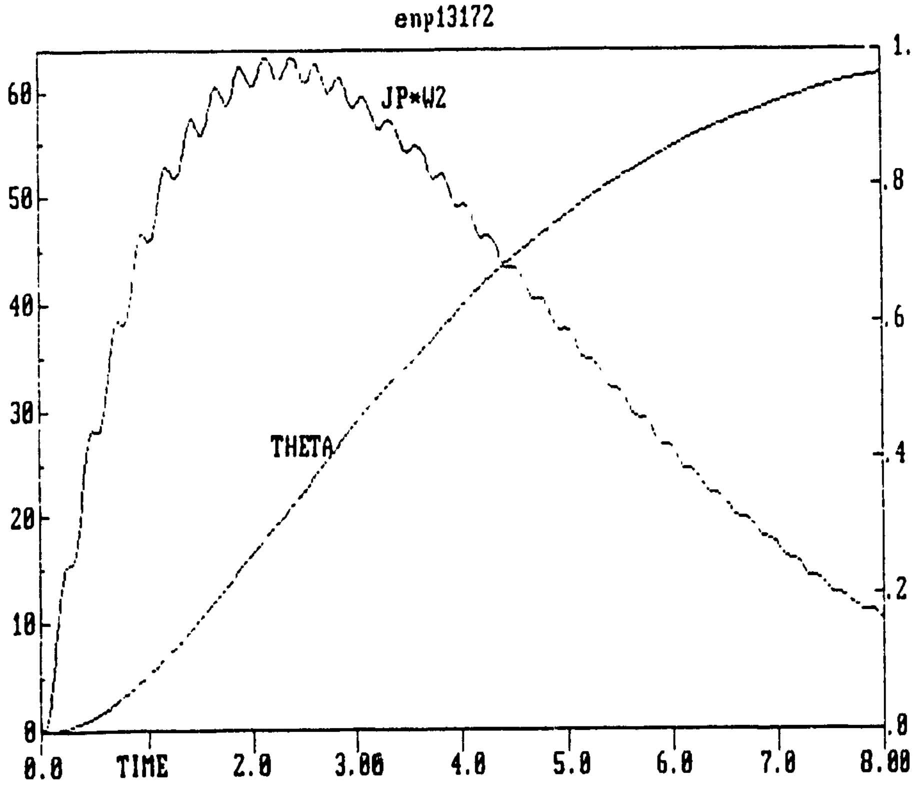


Fig. 4.5. Response of Radar Pedestal Unit Computed by CANVAS

4.2.2 One-Degree-of-Freedom Oscillator (M1_SDGRV)

OBJECTIVE

Validation of M1, SD, GRV components with analytically known solution (Phase III).

REMARKS

The 1 DOF spring-mass-damper system is the simplest model of vehicle suspension. Fig. 4.6 shows the CANVAS schematic for the system. The symbolic equations written by CANVAS are shown next to the schematic.

MODEL: M1_SDGRV

Component	Parameters	Units	Represents
M1	$m = 150$	kg	Quarter-vehicle body
	$v_0 = 0$	m/s	
GRV	$g = 9.81$	m/s^2	Gravity
SD	$C = .0001$	m/N	Suspension
	B = -deferred-	N·s/m	
GRD	-none-	—	Ground

ANALYSIS

Static deflection of the spring due to gravity is given by

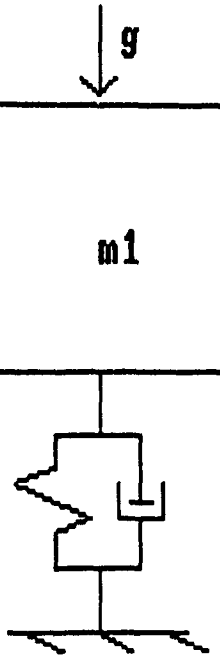
$$\Delta = m \cdot g \cdot C$$

Using the given parameters, $\Delta = .14715$ m.

Critical damping for the system occurs when

$$B = B_c = 2\sqrt{k \cdot m}$$

Using the given values of m and k, $B_c = 2449.48$ N·s/m.



```
f1 = Z(1) / 150               'm1
s2 = f1                       'm1
Zdot(3) = s2
f11 = f1                      'm1
f5 = f11                      'suspen.
s9 = 0                        'grd
f10 = (s9)                    'grd
f12 = f10                     'grd
f4 = f12                      'suspen.
f6 = (-f4 + f5)               'suspen.
f8 = f6                       'suspen.
Zdot(2) = f8                  'suspen.
e8 = Z(2) / .0001             'suspen.
f7 = f6                       'suspen.
e7 = f7 * 2449.48
e6 = (e7 + e8)                'suspen.
e5 = e6                       'suspen.
e11 = (-e5)                   'suspen.
s3 = 1471.5                   'grv
e13 = (s3)                    'grv
e1 = (-e13 + e11)             'm1
Zdot(1) = e1                  'm1
'M1_SDGRV
```

Fig. 4.6. CANVAS Schematic of 1 DOF Oscillator, with Equations

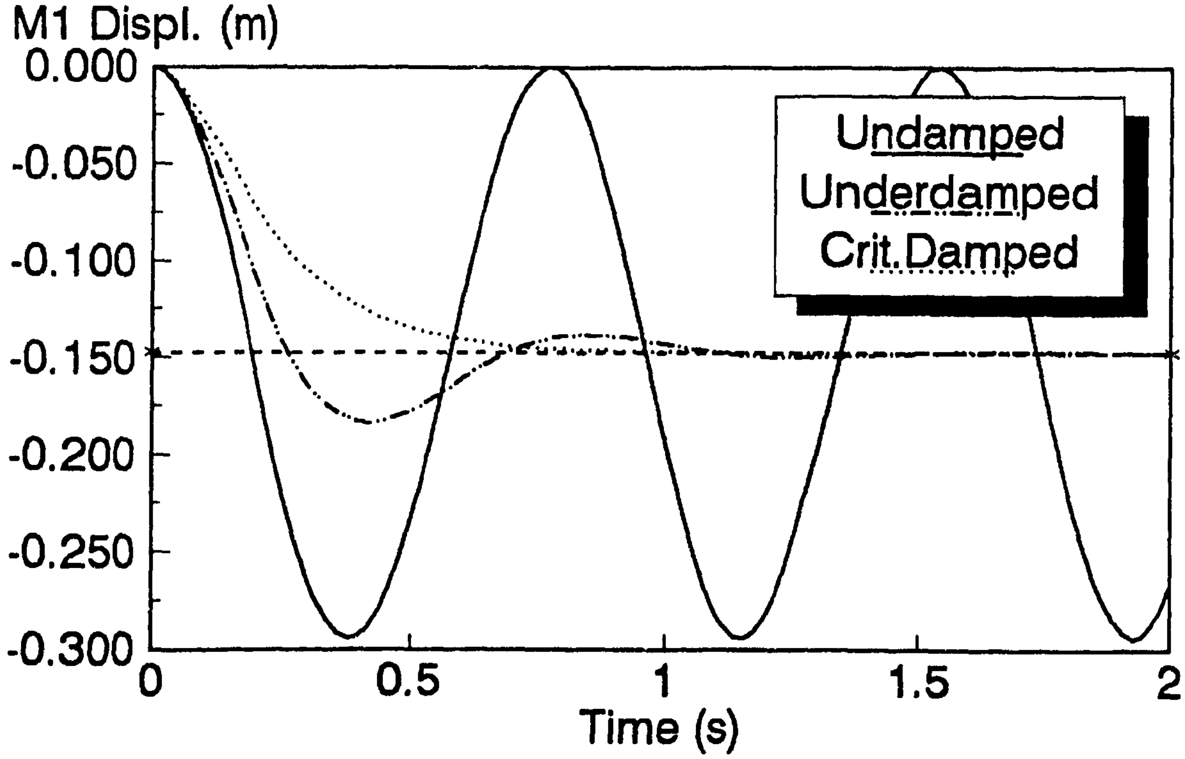


Fig. 4.7. Displacement Response of 1 DOF Oscillator for Varying Damping Parameter, B

RESULTS

Inspection of the dynamical equations shows that these are the classical equations of motion (in first order form) for a 1 DOF spring-mass-damper system.

Simulations were run for B = 0, B = 1000 N·s/m and $B=B_c$. For the zero-damping case, simple harmonic motion results, with period given by

$$T_n = 2\Pi\sqrt{\frac{m}{k}} = 0.770 \text{ s}$$

For the underdamped case, the oscillations diminish, while critical damping suppresses the oscillations altogether.

Fig. 4.7 is a plot of the displacement time response of the system as it settles from zero initial conditions to equilibrium under the influence of gravity for the above three values of damping.

An indicator (broken) line placed in the displacement time response at the prediced value of Δ shows that the mass settles downward to this static deflection, as expected.

4.2.3 Hydro-Gas Strut (M1CY1FVL)

OBJECTIVE

Comparison of hydro-pneumatic suspension to linear oscillator. Validation of CY1, ACC, FVL components.

REMARKS

Compared to the preceding example, we replace the linear spring with a hydro-gas strut. This is illustrated in the CANVAS schematic of Fig. 4.8. The presence of a gas-charged accumulator makes the force-displacement characteristic of the oscillator nonlinear. Essentially, as the accumulator volume is decreased, stiffness increases.

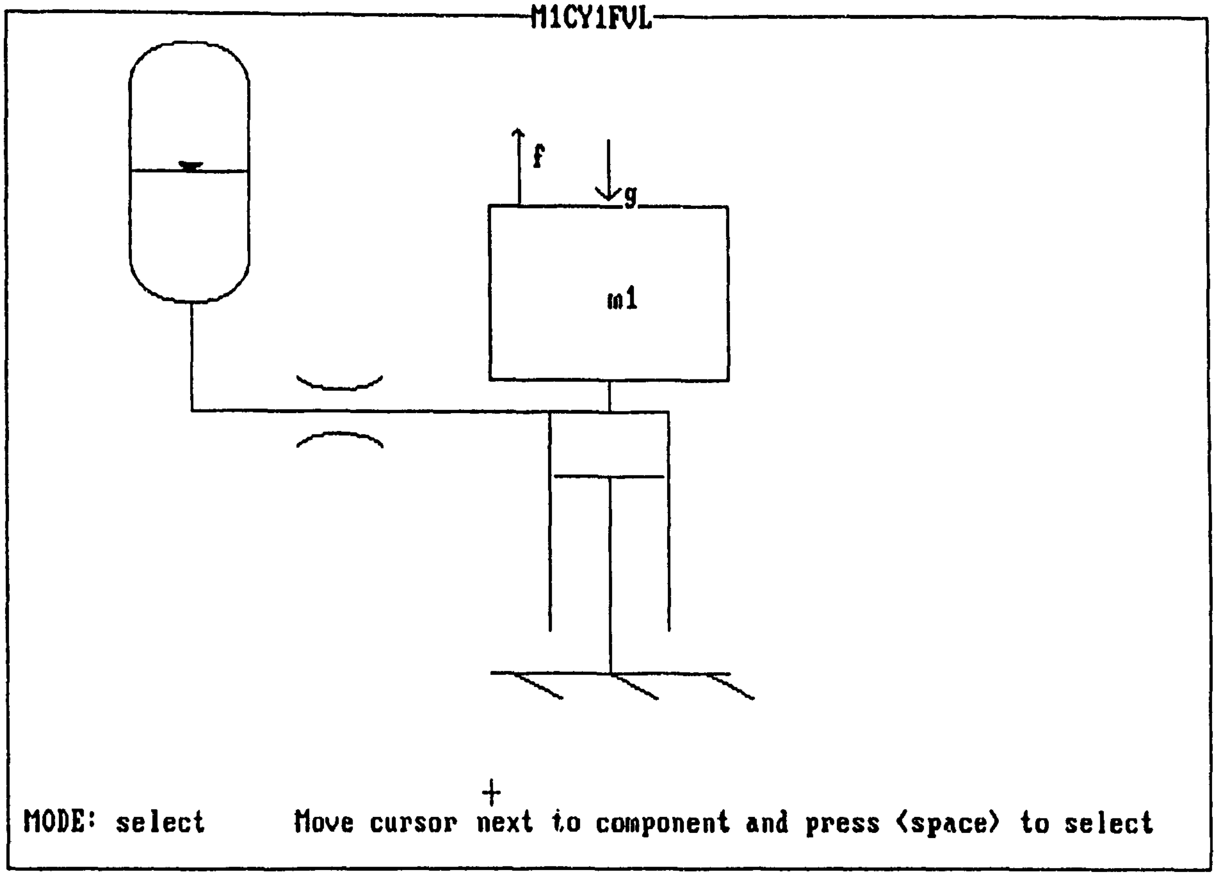
MODEL: M1CY1FVL

Component	Parameters	Units	Represents
M1	$m = 150$	kg	Inertia load
	$v_0 = 0$	m/s	
GRV	$g = 9.81$	m/s^2	Gravity
GRD	-none-	–	Ground
CY1	$A = 0.004905$	m^2	Hydro-gas strut
ACC	$P_0 = 3.0\times10^5$	Pa	
	$V_0 = 4.27\times10^{-4}$	m^3	
	$n = 1.4$	-	
FVL	R = -deferred-	$\left(\frac{m^7}{kg}\right)^{1/2}$	

ANALYSIS

The equivalent stiffness of the hydro-gas strut is given by

$$k_{eq} = \frac{dF}{dx} = \frac{nA^2P}{V}$$



M1CY1FVL

MODE: select   Move cursor next to component and press <space> to select

Fig. 4.8. CANVAS Schematic of Hydro-Gas Strut

For the parameter values used this yields a free-period

$$T_n = 2\Pi\sqrt{\frac{m}{k_{eq}}} = 0.5s$$

RESULTS

Examination of the response plot in Fig. 4.9 shows that the simulated solution has approximately the correct period. It also evidences the stiffening characteristic of the gas spring - the slope of the momentum time response is greater for that portion of the oscillation cycle when a positive volume inflow exists in the cylinder. The phase-plane plot in Fig. 4.10 evidences the system's non-linearity. A linear oscillator results in an elliptical shape for the closed curve. The stiffening effect of the strut is visible in the appreciable flattening of the negative displacement portion of the ellipse.

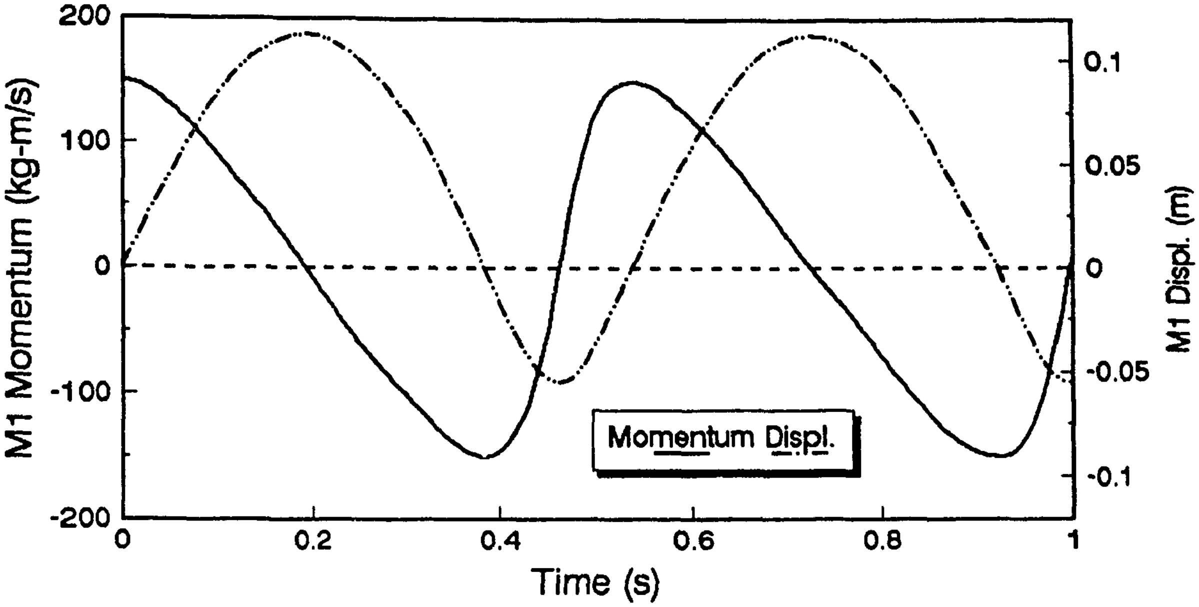


Fig. 4.9. Response of Hydro-Gas Strut for Non-Zero Initial Momentum

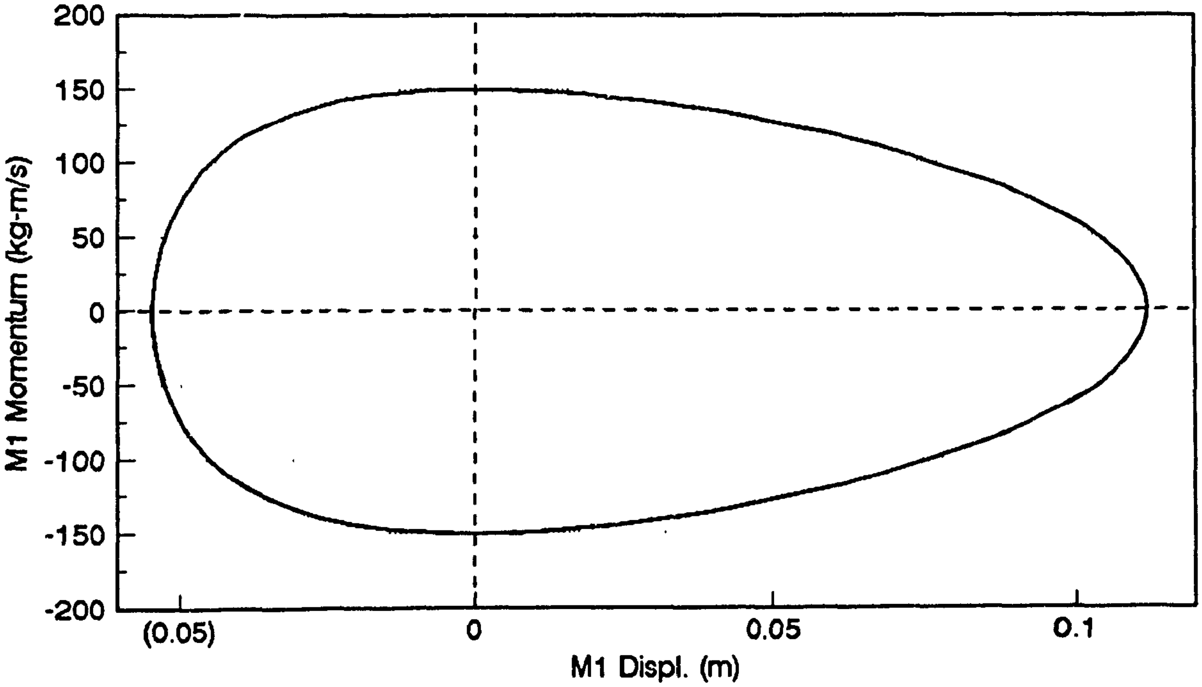


Fig. 4.10. Phase-Plane Plot of Hydro-Gas Strut Response

4.2.4 Open-Loop Fluid Power System (OPENLOOP)

OBJECTIVE

To illustrate the operation of the control valve (CV1) component coupled with a simple hydraulic system. Validate phase III.

REMARKS

As the name indicates, this is an open loop control system. Fluid flow from the control valve into the accumulator raises the load. Here the hydraulic supply to the valve is assumed to have constant pressure in order to simplify the model. When studying the power supply this assumption needs to be amended. Note that the constant supply pressure is represented graphically by the reservoir symbol, see Fig. 4.11.

MODEL: OPENLOOP

Component	Parameters	Units	Represents
M1	$m = 150$	kg	Load
	$v_0 = 0$	m/s	
GRV	$g = 9.81$	m/s^2	Gravity
CY1	$A_h = .001963$	m^2	Actuator
ACC	$P_0 = 749617.9$	Pa	Accumulator
	$V_0 = 1.186\times10^{-4}$	m^3	
	$n = 1.4$	–	
FVL	$R = 4\times10^{-7}$	$\left(\frac{m^7}{kg}\right)^{1/2}$	Damping restrictor
CV1	$R_{sl} = 1\times10^{-4}$		Proportional valve
	$R_{le} = 1\times10^{-4}$		
	$X_{min} = -\infty$	m	
	$X_{max} = +\infty$	m	
RSV.s	$P_s = 1.4\times10^6$	Pa	Power Supply
RSV.e	$P_e = 101300$	Pa	Tank (atm. pressure)

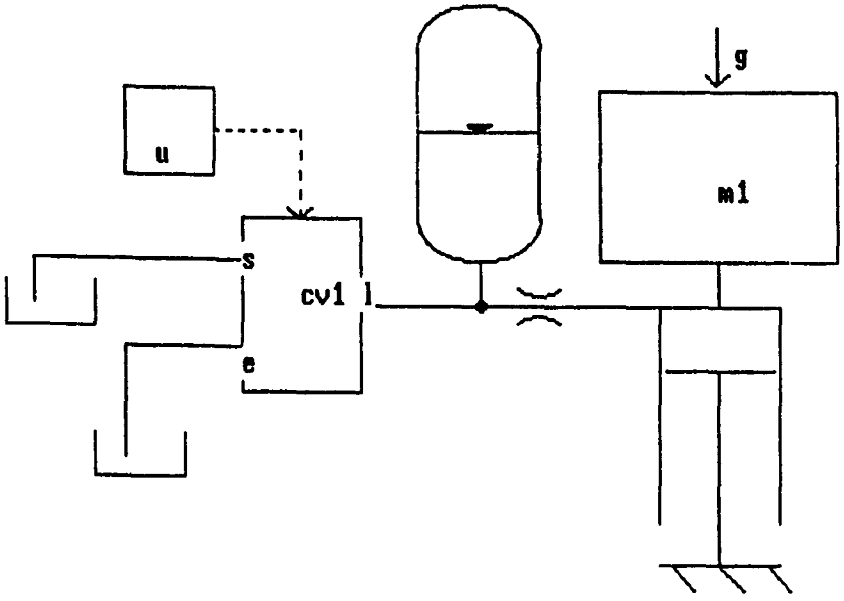


Fig. 4.11. CANVAS Schematic of Open-Loop Control System

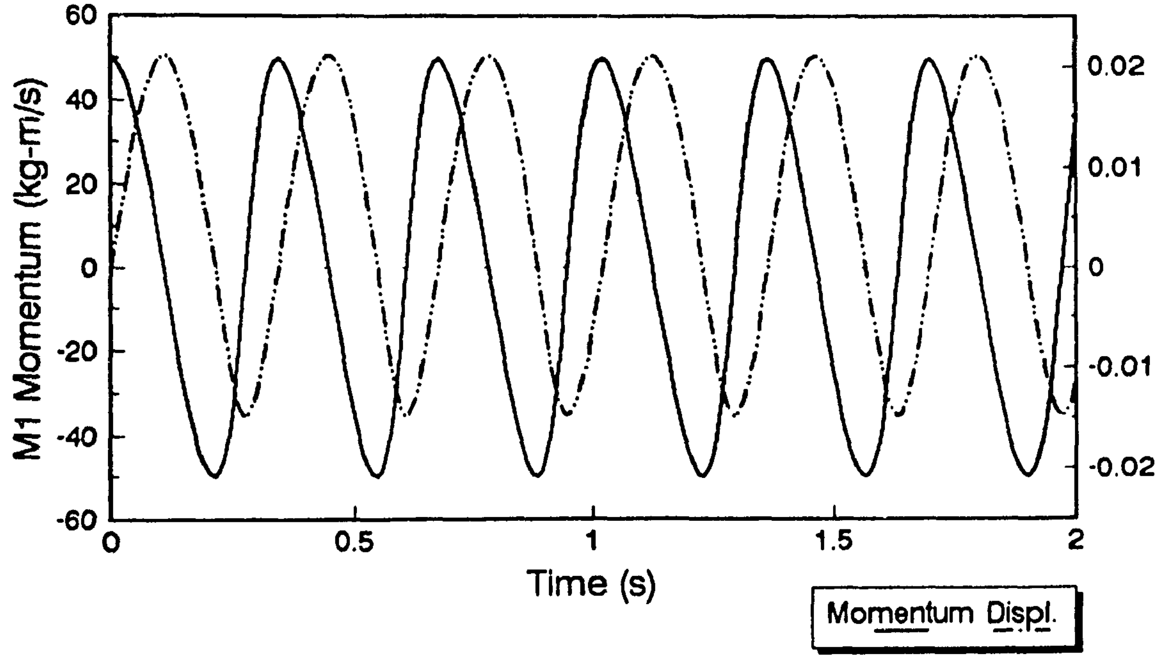


Fig. 4.12. Response of Open-Loop Control System for Initial Condition: Momentum = 50 Kg-m/s

ANALYSIS

Static balance was used to obtain precharge pressure given the cylinder area

$$P_0 \cdot A_h = m \cdot g$$

Supply pressure, P_s was selected based on the need to obtain equal valve gains on either side of null. With a symmetrical valve ($R_{sl} = R_{le}$) we require

$$P_s - P_0 = P_0 - P_e,$$

or

$$P_s = 2 \cdot P_0 - P_e \simeq 1.4 \times 10^6$$

Natural frequency is determined to be $f_n = 3.00$ Hz.

INPUTS

The input to the system is applied directly to the valve spool. The chosen input is a positive-going pulse of duration 0.5s and amplitude 0.001m followed by a negative-going pulse of equal duration and amplitude.

RESULTS

Natural behaviour

With no damping, no valve stroke and an initial momentum of 50 kg·m/s, the system behaves in a manner very similar to the hydro-gas strut of the preceding example: it oscillates with the predicted natural frequency - Fig. 4.12.

Command input

As expected, an opening of the valve causes a rise in the cylinder pressure; the load then accelerates upward, and oscillates with damping provided by the restrictor between the cylinder and the accumulator. Fig. 4.13 shows the time response for cylinder pressure, momentum of the load (M1) and load displacement. Note how steady-state pressure returns to the initial value, as expected.

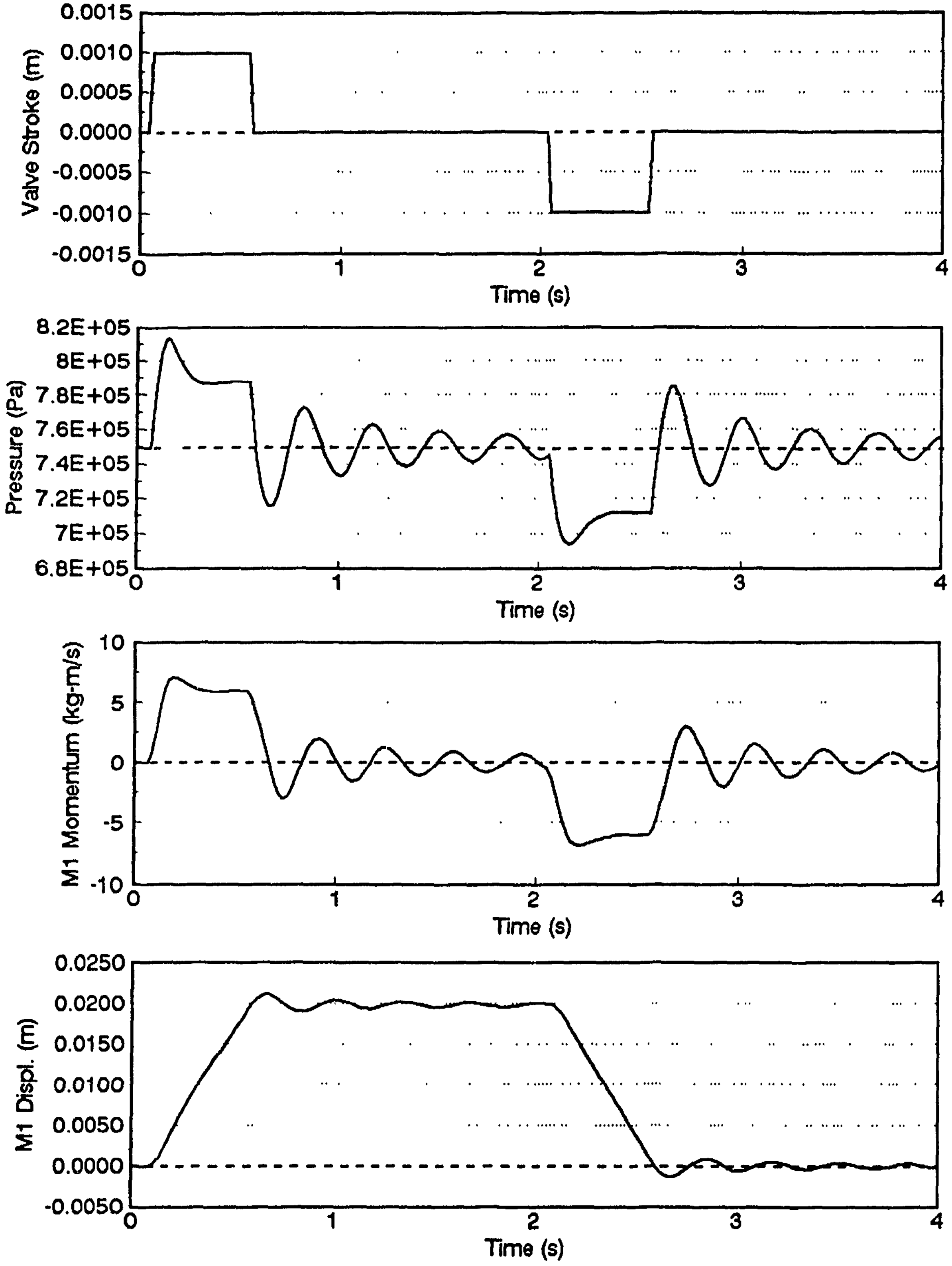


Fig. 4.13. Response of Open-Loop System to Valve Stroke Input

4.3 APPLICATION AND CORRELATION WITH PUBLISHED RESULTS

The examples appearing in this section may be distinguished from the preceding examples in that it is attempted herein to correlate CANVAS modelling and simulation with published studies in the vehicle domain. This constitutes more than *validation*. It shows how CANVAS may be used to carry out useful modelling and simulation work using a unified, graphically based methodology.

First we examine a quarter-vehicle model of a formula one car fitted with a hydro-gas strut and a feedback system to control ride height under the influence of aerodynamic downforce [44].

The next system under consideration is a roll-plane vehicle model of a crane, fitted with two different suspension schemes, as used in [10].

The concept of the active force generator is illustrated in an example which duplicates some of the results obtained by Thompson [45,46].

4.3.1 Formula One Race Car Active Suspension (DOMINY3)

OBJECTIVE

To reproduce (in part) the work of Dominy and Bulman [44], thus demonstrating CANVAS's modelling capability, from schematic to response. Validate phases II and III.

REMARKS

This example is based on Dominy and Bulman [44]. The objective of developing an active suspension for a formula one race car is to reconcile the need for extremely high suspension stiffness to static and quasi-static forces (e.g. aerodynamic load at speed) with the compliance necessary to attenuate terrain irregularities. Thus the specification of a limited-bandwidth load levelling system to counteract downforce, in combination with a relatively 'soft' suspension spring provided by a hydro-gas strut.

The schematic in Fig. 4.14 was developed to parallel the system described in [44]. It illustrates a highly simplified model of a vehicle as a 1 DOF (sprung mass only) system; it includes the components found in the open loop system described earlier, in addition to a feedback path between displacement measurement and valve stroke. This is seen to be equivalent to the control cable described in Dominy and Bulman.

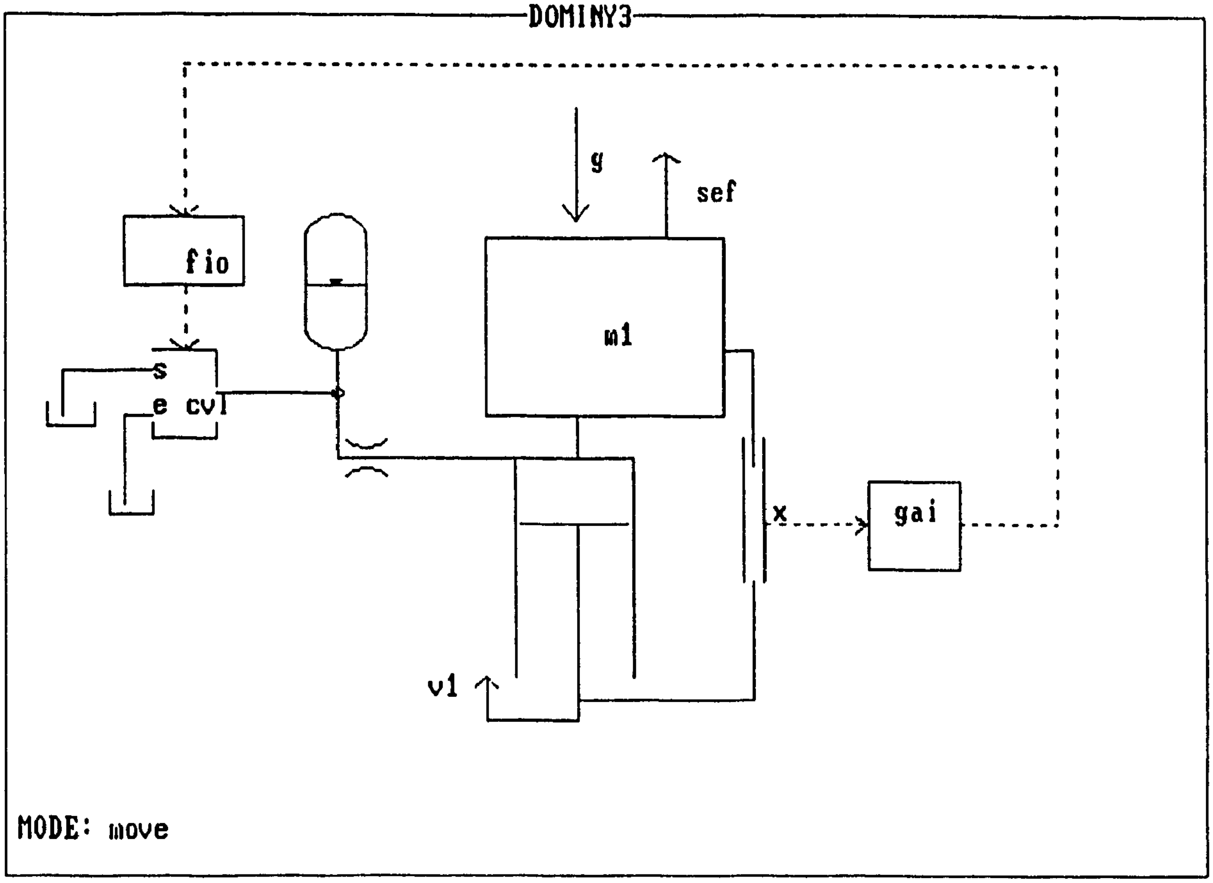


Fig. 4.14. CANVAS Schematic of "Formula One" Active Suspension

MODEL: DOMINY3

Component	Parameters	Units	Represents
M1	$m = 150$	kg	Quarter-Vehicle Mass
	$v_0 = 0$	m/s	
GRV	$g = 9.81$	m/s^2	Gravity
SEF	-deferred-	N	Aerodynamic force
SMO	-deferred-	m/s	Terrain input (vel.)
ACC	$P_0 = 749617.9$	Pa	Hydro-gas strut
	$V_0 = \left\{ \begin{array}{l} 1.186\times10^{-4} \\ 7.59\times10^{-5} \end{array} \right.$	m^3	
	$n = 1.4$	—	
CY1	$A_h = .001963$	m^2	
FVL	$R = 4\times10^{-6}$	$\left(\frac{m^7}{kg}\right)^{1/2}$	
CV1	$R_{sl} = 10^{-4}$		Control valve
	$R_{le} = 10^{-4}$		
	$X_{min} = -\infty$	m	
	$X_{max} = +\infty$	m	
LDT	-none-	—	Rel.Displ.Transducer
FIO	$\tau = .03$	s	Control valve dynamics
GAI	G = -deferred-	—	Feedback gain
RSV.s	$P_s = 7.5\times10^6$	Pa	Power Supply
RSV.e	$P_e = 101300$	Pa	Tank pressure (1atm)

ANALYSIS

Dominy and Bulman define accumulator precharge pressure (P_0') and free volume (V_0') that are not in equilibrium with gravity. They must therefore let the system settle to static equilibrium before simulation. We have found the equivalent free volume and precharge pressure. These are summarized in Table 4.1.

Equilibrium precharge pressure is found from

$$P_0 \cdot A_h = m \cdot g$$

Static deflection Δx for the passive system (Gain set to 0) under load can be predicted by equating the force produced by the strut to the applied load

$$A_h \cdot P_g = F_a + m \cdot g$$

where

$$P_g = P_0 \cdot \left(\frac{V_0}{V_0 + A \cdot x} \right)^n$$

For $V_0 = 1.186 \times 10^{-4}$ and a load of $F_a = 2000$ N, the expected downward deflection is $x = .0277$ m. With equal load, but a smaller inital volume of $V_0 = 7.59 \times 10^{-5}$, the static deflection decreases to $x = .0177$, reflecting a 'stiffer' suspension.

TABLE 4.1 - Hydro-Gas Suspension Characterstic

P'_0	V'_0	V_0	k_{eq}	f_n
3.0×10^5	.00050	2.599×10^{-4}	15600	1.62
3.0×10^5	.00032	1.674×10^{-4}	24300	2.03
1.0×10^5	.00050	1.186×10^{-4}	34100	2.40
1.0×10^5	.00032	7.590×10^{-5}	53200	3.00

Fig. 4.15 shows the Force-Displacement characteristic for the hydro-gas strut, assuming the given cylinder effective area. The stiffening characteristic is evident from the increasing slope at the greater deflections.

The equivalent stiffness and natural frequency of the hydro-gas strut supporting a mass are shown in Table 4.1 for several values of V'_0 and P'_0, as used in [44].

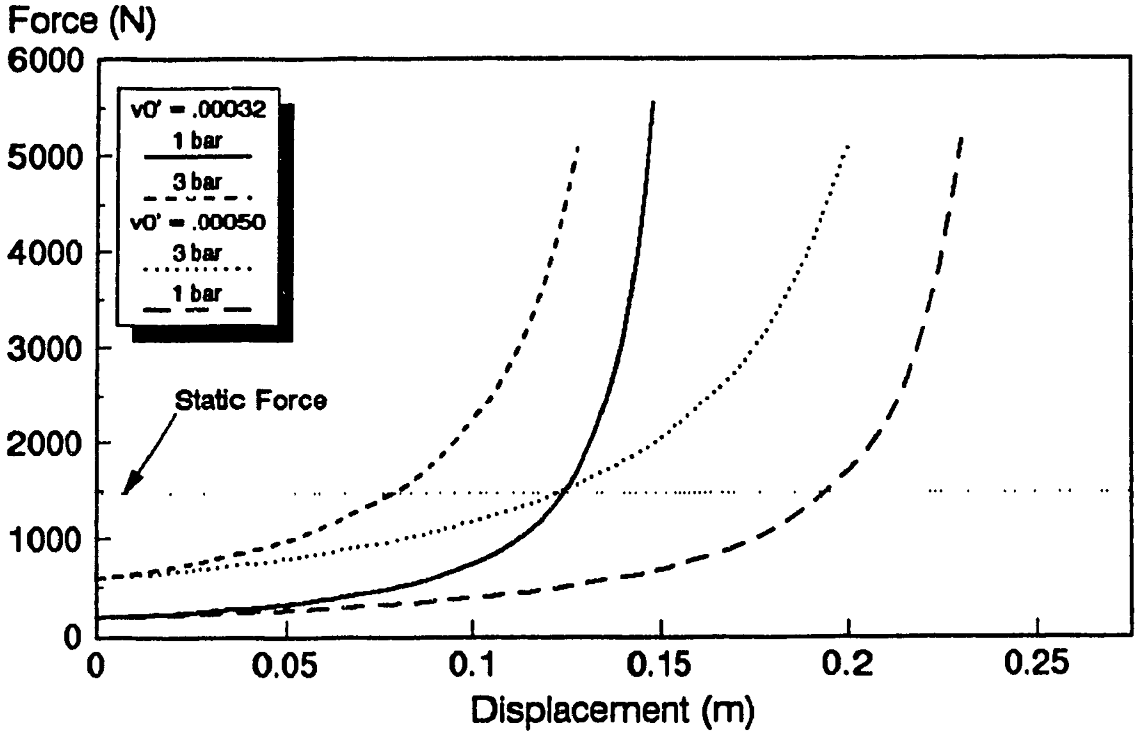


Fig. 4.15. Hydro-Gas Spring Characteristic

First-order valve dynamics as given in [44] are a result of spring/damper acting on the massless spool. This is represented in the CANVAS schematic by the FIO block. We have empirically determined an appropriate value of the valve time constant; sensitivity to this parameter is low.

INPUTS

Excitation to the vehicle is provided by either a force source (SEF) or a motion source (terrain input). These are shown as part of the schematic; the exact form of the input is specified at the time a simulation run is made. For our simulations these cases were examined:

- Downward force acting on the vehicle body, and increasing linearly to a maximum of 2000N in 0.5s. This corresponds to an applied aerodynamic load, and we expect that the feedback in the suspension will compensate (in steady-state) for this load, shown in Fig. 4.16.
- 'white noise' force input acting on vehicle body, to evidence dominant system modes through frequency analysis.
- Single half-sine bump of wavelength 5m traversed at a speed of 30 m/s. Here the active suspension is not expected to react significantly.

RESULTS

Static deflection

Examination of the plots in Figs. 4.17 and 4.18, shows that the center of mass of the passive system indeed settles downward by the amount predicted analytically for the given load of 2000 N.

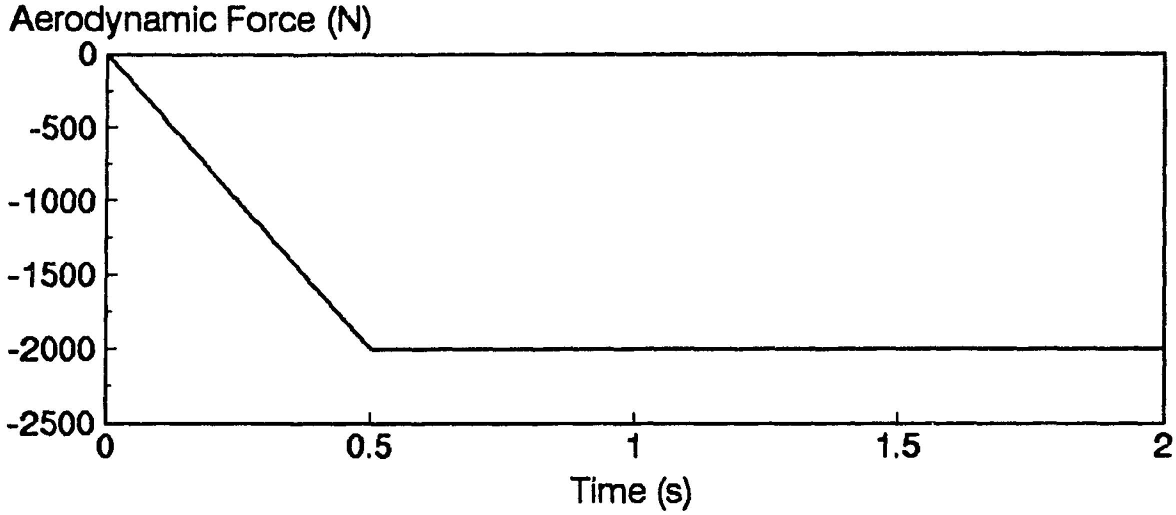


Fig. 4.16. Aerodynamic Input for "Formula One" Example

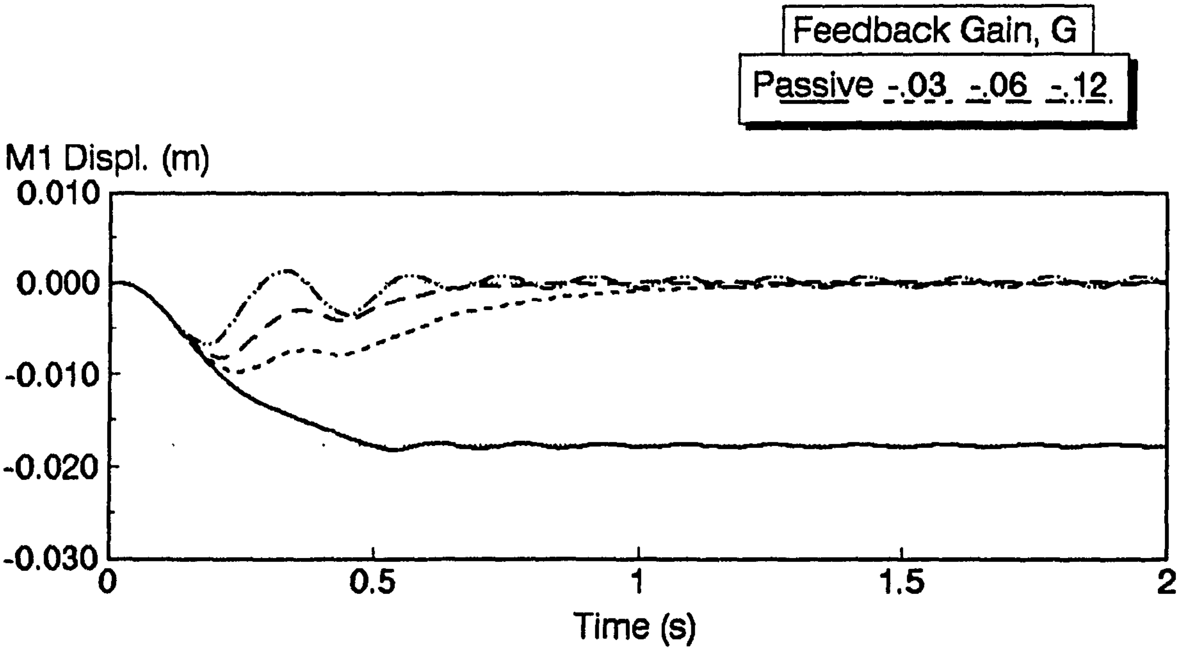


Fig. 4.17. Displacement Response of "stiff" Active Suspension

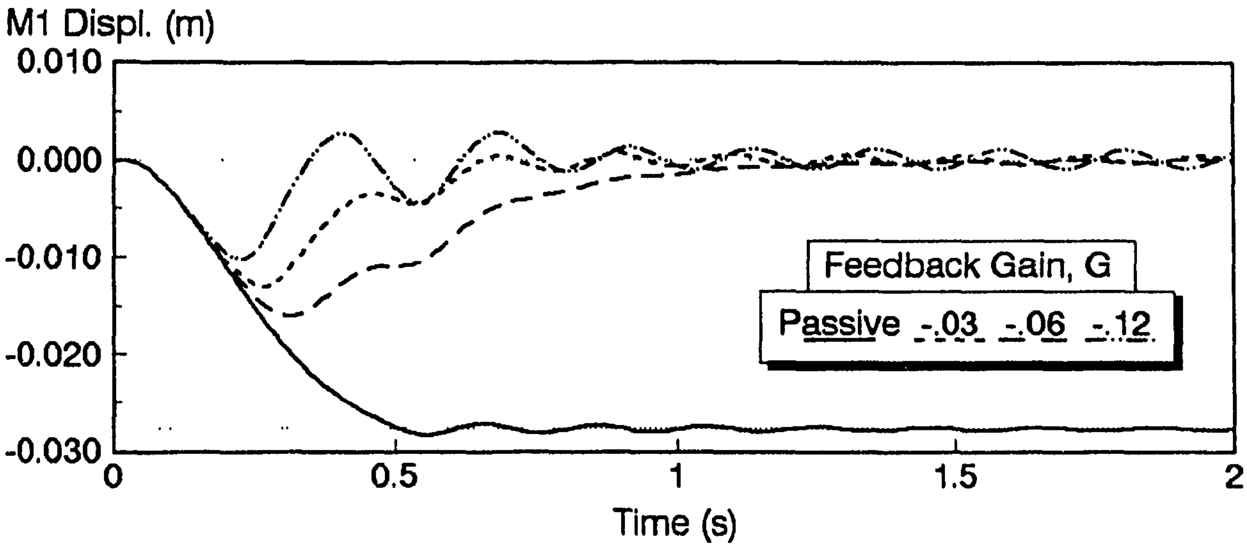


Fig. 4.18. Displacement Response of "soft" Active Suspension

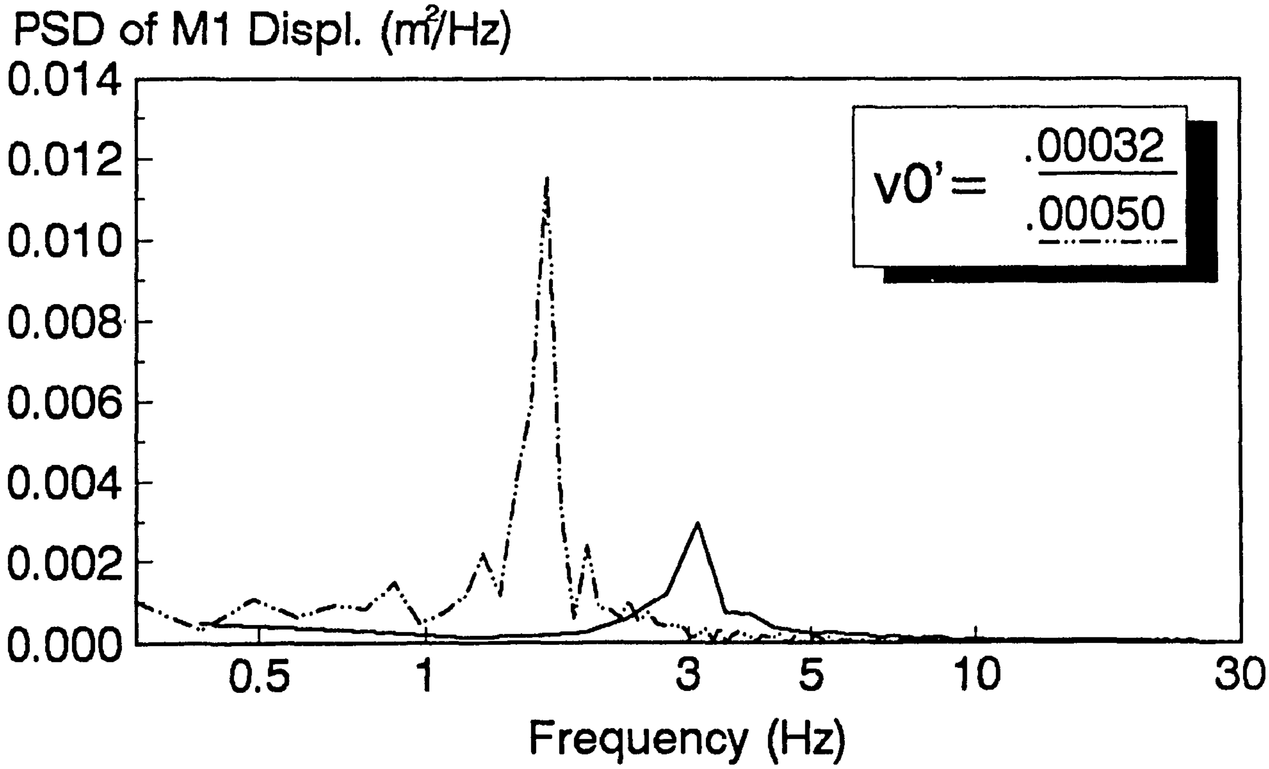


Fig. 4.19. Frequency Response of "Formula One" Active Suspension

Natural frequency

In order to confirm the model's validity with respect to analytical natural frequency, the following procedure was used: with valve gain set to 0 (passive system), the vehicle body was excited by a gaussian white noise force input; the displacement response was subsequently frequency analyzed (using the FFT feature of POST), to yield the plots in Fig. 4.19. Comparison of the plots with Table 4.1 shows that the frequency of peak response amplitude corresponds to the analytical natural frequency.

Correlation with published results

Examination of the dynamical equations in Fig. 4.20 reveals their similarity with those derived manually in [44]. Simulation plots for various inputs are shown in Figs. 4.17-4.19 and 4.21. They reveal the same trends as in the original paper, namely:

- Displacement feedback eliminates steady-state deflection under load. Self-levelling action becomes quicker as feedback gain is increased (Figs. 4.17 and 4.18).
- Initial response to a single bump is very similar for the passive and active system, showing that the active system behaves as a soft suspension for rapid road inputs. The effect of integral-displacement feedback is evident through more persistent oscillations (Fig. 4.21).

A lack of complete parameter information in the reference precludes a perfect match in the simulation results; the results obtained, however, agree *qualitatively* with those published. This example does not call into question the validity of the original model, but rather demonstrates the ability of the CANVAS software to replicate the results of a published paper automatically, with a schematic drawing as starting point for the model.

File = DOMINY3.INI

```
'DOMINY3
CONST F$ = "DOMINY3"
nState = 5
nOutputs = 2
nXCT = 2
xctmean$(1) = "Velocity"
xctmean$(2) = "FORCE"
Z0(1) = 0: m$(1) = "M1.momentum"
Z0(2) = 0: m$(2) = "ACC.tot_inflow"
Z0(3) = 0: m$(3) = "FIO"
Z0(4) = 0: m$(4) = "M1.disp"
Z0(5) = 0: m$(5) = "LVD"
m$(6) = "vel. input"
m$(7) = "force input"
```

File = DOMINY3.OUT

```
s40 = Z(5)                          'lvd  INT 31
s42 = d(4) * (s40)                  'gai  GAI 32
s24 = (s42)                         'fio  GAI 35
s25 = Z(3)                          'fio  INT 34
s23 = 1 / d(3) * (-s25 + s24)       'fio  ATT 33
Zdot(3) = s23
s12 = xct(1, t)                     'smo  XCT 16
f32 = (s12)                         'smo  SF 17
s38 = f32                           'cyl Vram 1 8
f2 = Z(1) / 150                     'm1  I 1
s39 = f2                            'm1  1 2
Zdot(5) = s39 - s38
s1 = f2                             'm1  1 2
Zdot(4) = s1
s41 = Z(3)                          'fio  INT 34
Temp = (s41)                        'cv1 actual displ. LIM 22
IF Temp < -1 THEN Temp = -1
IF Temp > 1 THEN Temp = 1
s15 = Temp
s28 = 7500000!                      'rsv  KON 39
e22 = (s28)                         'rsv  SE 40
'acc  C:polytropic 15
e36 = 749617.9 * (d(2) / (d(2) - Z(2))) ^ 1.4
e37 = e36                           'jun Junction 0 30
e20 = e37                           'cv1 load press 0 24
e21 = (-e20 + e22)                  'cv1  1 27
'cv1 supply-load flow MSR+ 28
f21 = -(s15 > 0) * s15 * .0001 * SGN(e21) * SQR(ABS(e21))
f20 = f21                           'cv1  1 27
s16 = Temp
e19 = e37                           'cv1 load press 0 24
s26 = 101300                        'rsv  KON 36
e27 = (s26)                         'rsv  SE 37
e17 = e27                           'rsv  0 38
```

Fig. 4.20. CANVAS Equation Set for "Formula One" Active Suspension

File = DOMINY3.OUT (cont'd)

```
 e18 = (-e17 + e19)                'cvl  1 26
'cvl load-exhaust flow MSR- 29
 f18 = -(s16 < 0) * s16 * .0001 * SGN(e18) * SQR(ABS(e18))
 f19 = f18                         'cvl  1 26
 f37 = (-f19 + f20)                'cvl load press 0 24
 f5 = f32                          'cyl Vram 1 8
 f31 = f2                          'm1  1 2
 f6 = (-f31 + f5)                  'cyl  0 5
 f7 = f6 * .001963                 'cyl  TF 6
 f10 = (-f7)                       'fvl  0 13
 f9 = f10                          'fvl  1 11
 f35 = (-f9)                       'fvl  0 12
 f36 = (f35 + f37)                 'jun Junction 0 30
 Zdot(2) = f36                     'acc  C:polytropic 15
 s14 = xct(2, t)                   'sef  XCT 20
 e34 = (s14)                       'sef  SE 21
 s13 = 1471.5                      'grv  KON 19
 e33 = (s13)                       'grv  SE 18
 e35 = e36                         'jun Junction 0 30
 e9 = e35                          'fvl  0 12
 f8 = f10                          'fvl  1 11
 e8 = SGN(f8) * (f8 / d(1)) ^ 2    'fvl  RSQ 10
 e10 = (-e8 + e9)                  'fvl  1 11
 e7 = e10                          'fvl  0 13
 e6 = e7 * .001963                 'cyl  TF 6
 e31 = e6                          'cyl  0 5
 e2 = (e31 - e33 + e34)            'm1  1 2
 Zdot(1) = e2                      'm1  I 1
 x(6) = s12                        'vel_input
 x(7) = s14                        'force_input
 'DOMINY3
```

Fig. 4.20. CANVAS Equation Set for "Formula One"
Active Suspension

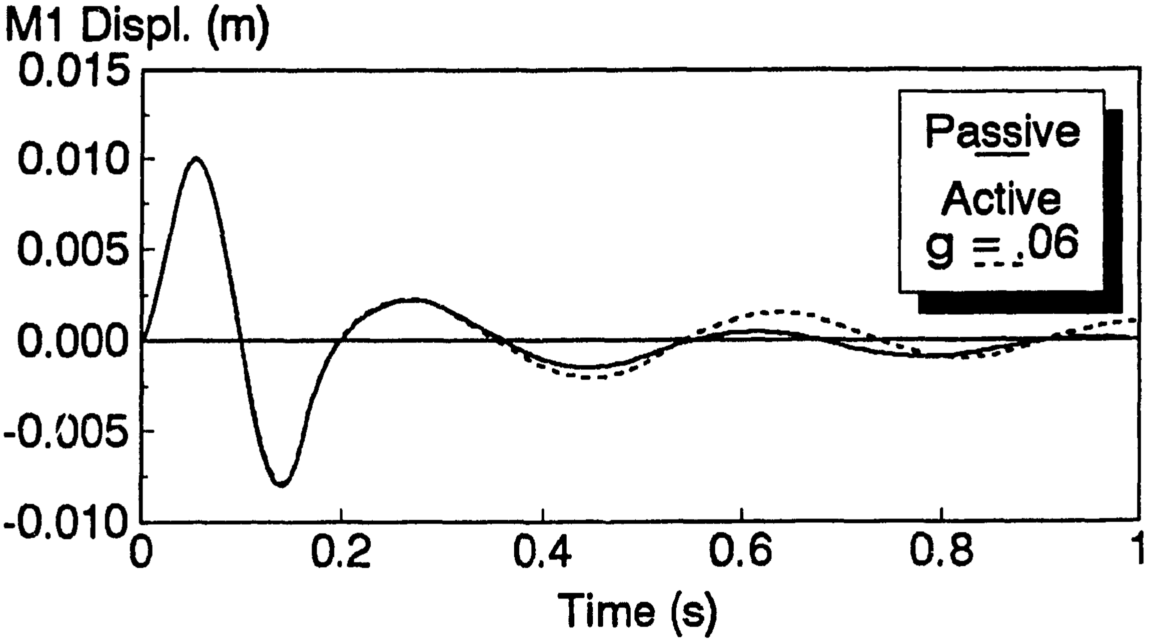


Fig. 4.21. Comparison of Passive and Active Suspension Response to Half-Sine Input

4.3.2 Hydraulic Suspension for Crane Vehicle

OBJECTIVE

Paralleling the work of Felez and Vera [10], develop two models of Crane Vehicle with Hydraulic Suspension. Validate phases I, II and III.

REMARKS

Felez and Vera [10] use bond graphs to develop a modular set of vehicle and suspension models of a crane. The roll-plane vehicle model is augmented in turn with each of three hydraulic suspension systems:

- independent cylinder
- linked cylinder
- active

In this section we partially reproduce the work of [10] by developing models of the first two systems. The first may be considered the baseline system, while the linked cylinder suspension seeks to limit cornering roll by connecting the upper chamber of the left hand side hydraulic cylinder with the lower chamber of the right hand side one, and vice-versa.

It should be noted that the methodology followed in [10] entails manual creation of bond graphs in preparation for their entry into a program (BONDYN) which generates equations. This is in contrast with CANVAS's method of direct schematic entry. The model parameter information given in [10] is unfortunately incomplete. Furthermore, one of the figures is misleading, if not incorrect. For this reason, it was decided that an exact duplication of the results in that reference was not feasible. Hence, a qualitative agreement is sought.

The model used in [10] includes 2 rigid bodies of 2DOF, representing the crane body and a rigid axle. The CANVAS schematics for two such models are shown in Figs. 4.22 and 4.23. The former shows the independent suspension; the latter shows the linked-cylinder suspension. However, in this study it is found that axle dynamics are not of significant influence to the roll beviour of the vehicle. For this reason, we discard the axle altogether and work on a reduced vehicle model of 2 DOF only.

The bond graph structure for the full 4DOF model with linked-cylinder suspension was nonetheless obtained with CANVAS and is shown in Fig. 4.24. The schematics of the reduced models are given in Figs. 4.25 and 4.26 respectively. Note that the right side ground input (SMO - appearing as arrows) in the original schematics is replaced in one of the reduced schematic diagrams by a fixed ground component (GRD), as this suffices for simulation.

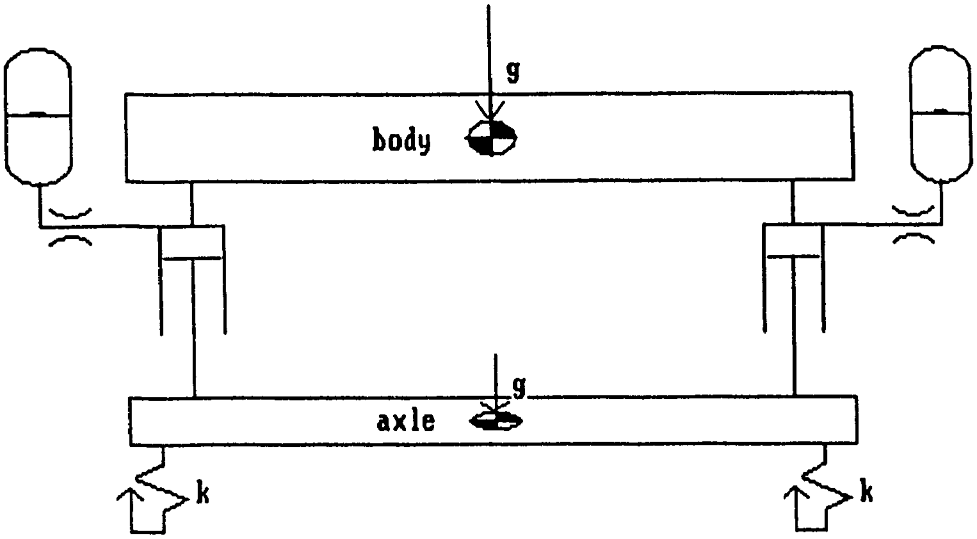


Fig. 4.22. Two-Body (4DOF) Roll-Plane Model of Crane Vehicle with Independent-Cylinder Suspension

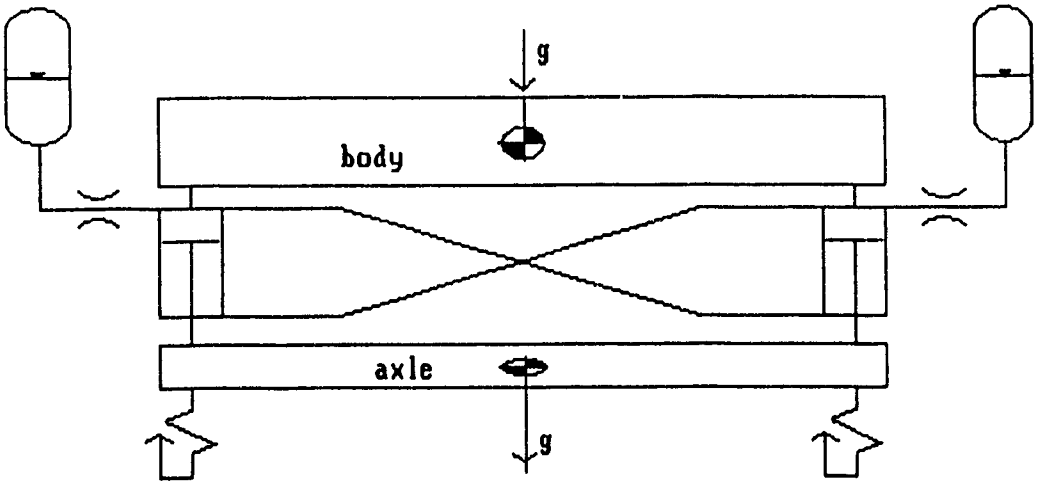


Fig. 4.23. Two-Body (4DOF) Roll-Plane Model of Crane Vehicle with Linked-Cylinder Suspension

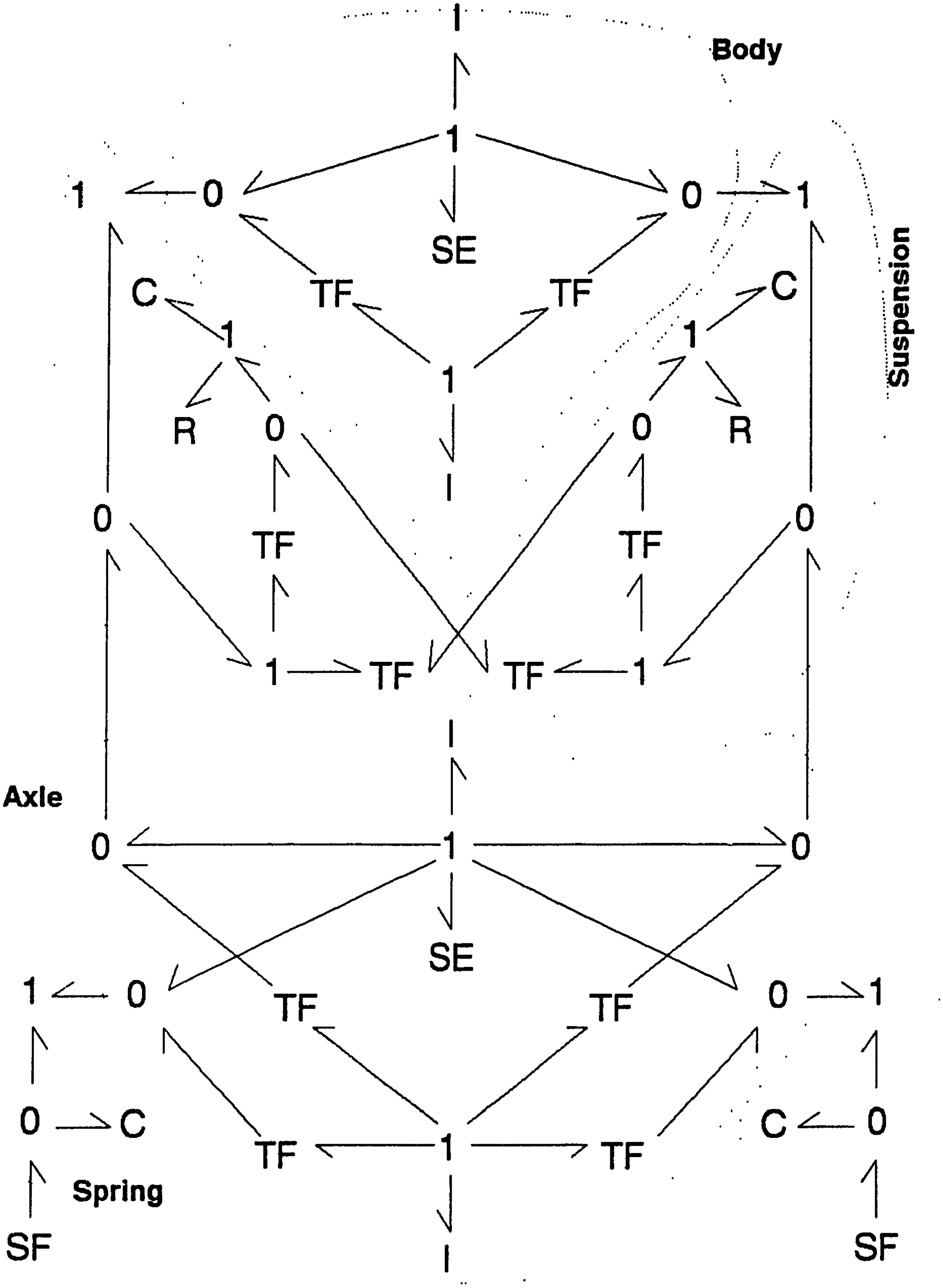


Fig. 4.24. Bond Graph Structure Derived by CANVAS for the 4 DOF Linked-Cylinder Suspension Crane Model

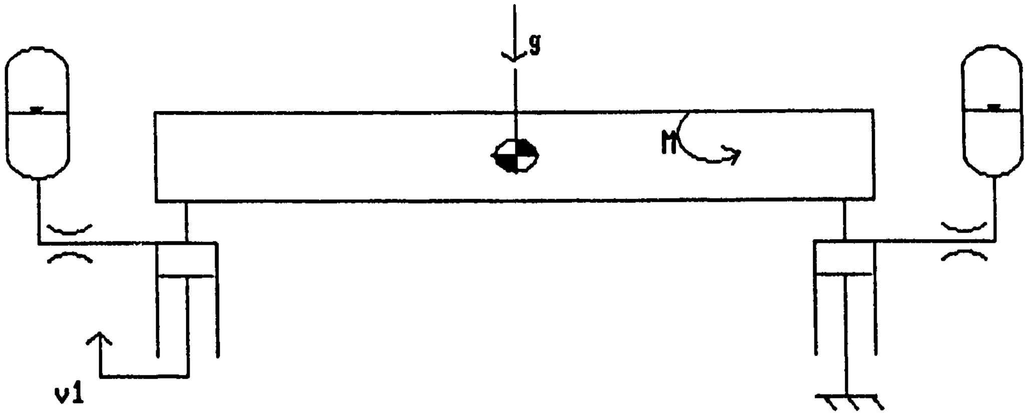
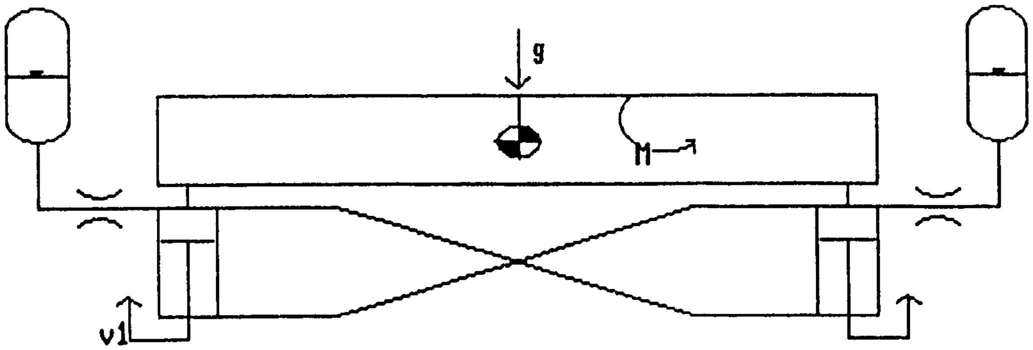


Fig. 4.25. Simplified Schematic of Independent-Cylinder Model



4.26. Simplified Schematic for Linked-Cylinder Model

MODEL: M2_CY1_I (Independent cylinder suspension)

Component	Parameters	Units	Represents
M2	$m = 1200$	kg	Mass
	$v_0 = 0$	m/s	
	$J = 1350$	$kg \cdot m^2$	Rotational inertia
	$\omega_0 = 0$	rad/s	
	$a_1 = -.575,\ a_2 = .575$	m	Attachment points
GRV	$g = 9.81$	m/s^2	Gravity
ACC	$P_0 = 606714$	Pa	Hydro-gas strut
	$V_0 = .0015$	m^3	
	$n = 1.4$	—	
CY1	$A_h = 9.701 \times 10^{-3}$	m^2	
FVL	R = -deferred-		

MODEL: FAGAN1 (Linked cylinder suspension)

Component	Parameters	Units	Represents
M2	$m = 1200$	kg	Mass
	$v_0 = 0$	m/s	
	$J = 1350$	$kg \cdot m^2$	Rotational inertia
	$\omega_0 = 0$	rad/s	
	$a_1 = -.575,\ a_2 = .575$	m	Attachment points
GRV	$g = 9.81$	m/s^2	Gravity
ACC	$P_0 = 606714$	Pa	Hydro-gas strut
	$V_0 = .0015$	m^3	
	$n = 1.4$	—	
CY3	$A_{net} = 9.701 \times 10^{-3}$	m^2	
	r = -deferred-	—	
FVL	R = -deferred-		

ANALYSIS

Physical parameters given in [10] were analyzed and found incomplete/inadequate/inconsistent, since several key parameters are missing, while some which are given are not even necessary for the model. For instance, it is not specified what values were taken for the roll moments of inertia for both body and axle. Cylinder upper stroke is not ever used.

Therefore, design considerations were used to arrive at the above parameter values. For example, selecting the cylinder net area and accumulator precharge pressure, is accomplished as follows:

Specifications

Accumulator Volume	V	: $0.0015\ m^3$
Mass per axle	m	: 1200 Kg
Track between cylinders	$2a$	: 1.15 m
Bounce natural freq.	f_0	: 1.5 Hz

Design constraints

$$k_{eq} = 2\cdot \frac{n\cdot A_{net}^2\cdot P_0}{V_0} \qquad \text{(equivalent stiffness)}$$

$$m\cdot g = 2\cdot P_0\cdot A_{net} \qquad \text{(static equilibrium)}$$

Resulting parameters

Cylinder Net Area	A_{net}	: $9.701\times10^{-3} m^2$
Accumulator Precharge Press.	P_0	: 606713.8 Pa

INPUTS

Displacement step (asymmetrical road input)

Examination of Fig. 6.2 in [10] reveals that there is a steady-state chassis vertical displacement of .025 m. Thus, the velocity excitation at the left side (V_1) cannot be as shown in Fig. 5 of that reference. Rather, it is inferred that the actual profile is a displacement step of amplitude

.05 m and duration of 5 s.

Thus, in our simulation, the required displacement profile is replaced by an equivalent velocity profile (An upward pulse followed by a downward pulse, each with area equal to .05 m spaced apart by 5 s) - see Fig. 4.27.

Rolling moment

To simulate the rolling behaviour of the vehicle entering a curve, the authors of [10] consider a moment resulting from a centrifugal force acting at the c.g. (at a height, h, above the vehicle roll centre); see Fig. 4.28. They assume that the incremental moment due to the lateral c.g. shift is of interest. We suppress this assumption, as it is only of second-order effect for the small roll angles experienced by the crane vehicle. Thus, the roll moment due to curving is given by

$$M_r = m \cdot a_l \cdot h$$

where m is the vehicle mass, a_l is the assumed lateral acceleration, and h is the height of c.g. above the roll centre. For the simulations, a lateral acceleration, a_l = 1.04 $m/s^2$was assumed. Thus, assuming h = 0.8 m, and using the given mass, the corresponding rolling moment is M_r = 1000 N·m.

The rolling moment is apparent in the schematics of Figs. 4.26 and 4.27 as a semi-circular arrow labelled with an 'M' . For simulation, the applied moment is actually given by a rounded step rising to the maximum value of M_r, given above.

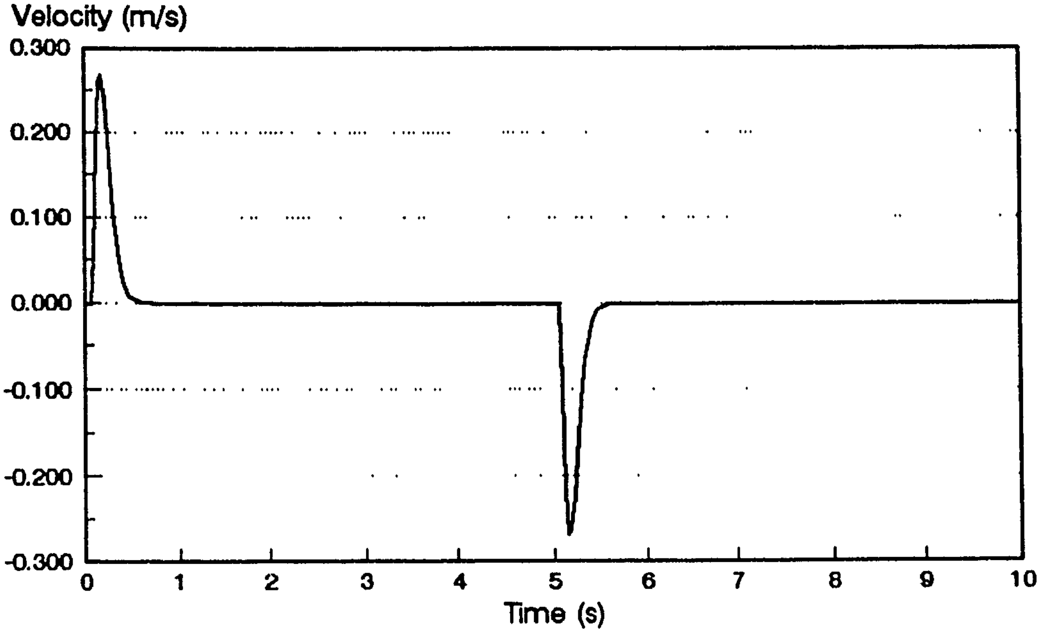


Fig. 4.27. Asymmetrical Road Velocity Input for Crane Vehicle

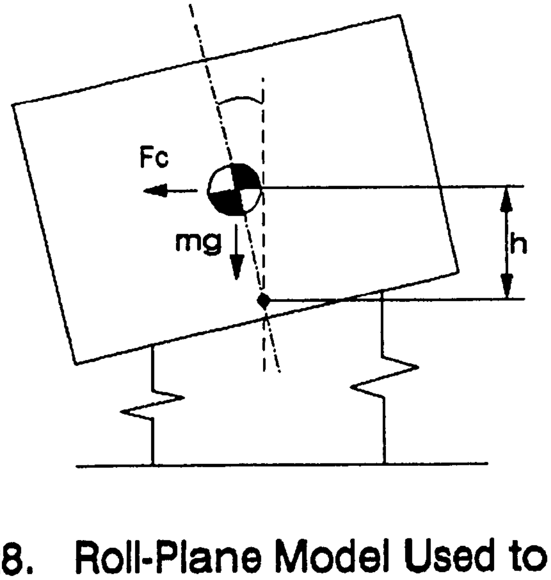


Fig. 4.28. Roll-Plane Model Used to Obtain Equivalent Cornering Moment

RESULTS

Simulations were carried out for both models, in three series.

- Bounce mode validation
- Asymmetrical road input
- Cornering behaviour

Series I: Bounce Mode

In the first series it is verified that the linked cylinder suspension bounce mode behaviour is equivalent to the behaviour of the independent cylinder suspension. Initial conditions for both models were set so that the vehicle has linear momentum corresponding to a downward velocity of 1 m/s. The response plots for models M2_CY1_I (independent) and FAGAN1 (linked) are shown respectively in Figs. 4.29 and 4.30; they are identical as expected.

Series II: Asymmetrical Road Input

In the second series, the asymmetrical road velocity input is used to bring out differences in the roll mode behaviour of the two suspension systems. It is expected that the linked cylinder suspension will show a reduced transient roll response to such an input. This is because the cross-link serves to equalize cylinder pressures on both sides of the vehicle.

Fig. 4.31 compares the response of the independent-cylinder suspension to that of a linked-cylinder suspension with two area ratio, $r = A_h/A_r = 2$; it shows that the linked cylinder suspension achieves a considerable reduction in the transienc angular displacement, as well as in chassis vertical displacement.

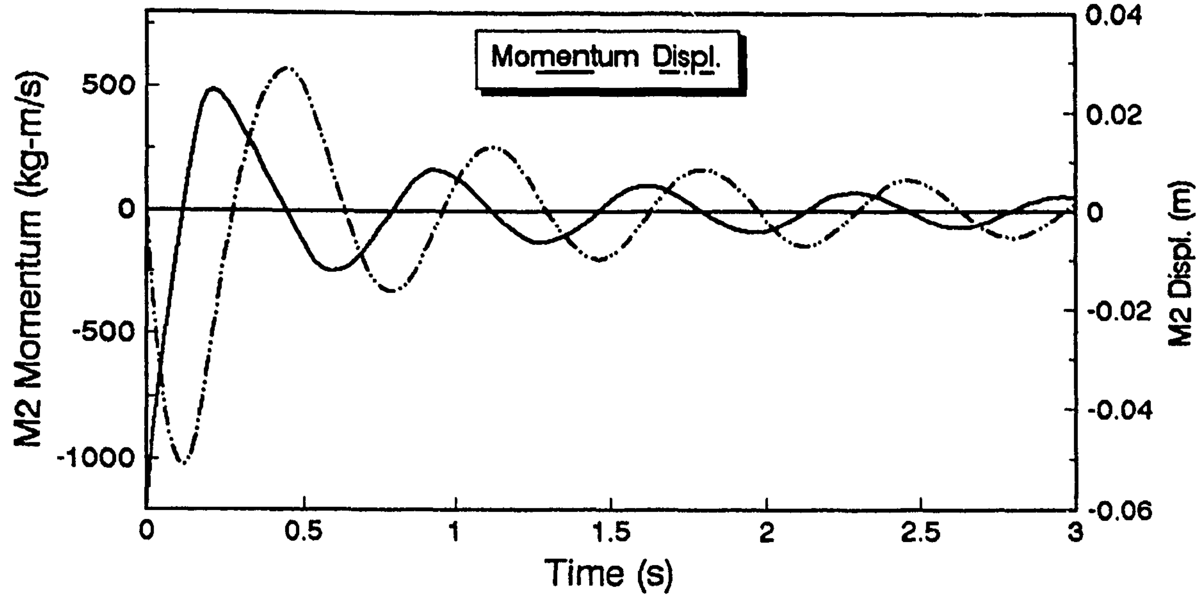


Fig. 4.29. Bounce Response of Independent-Cylinder Model

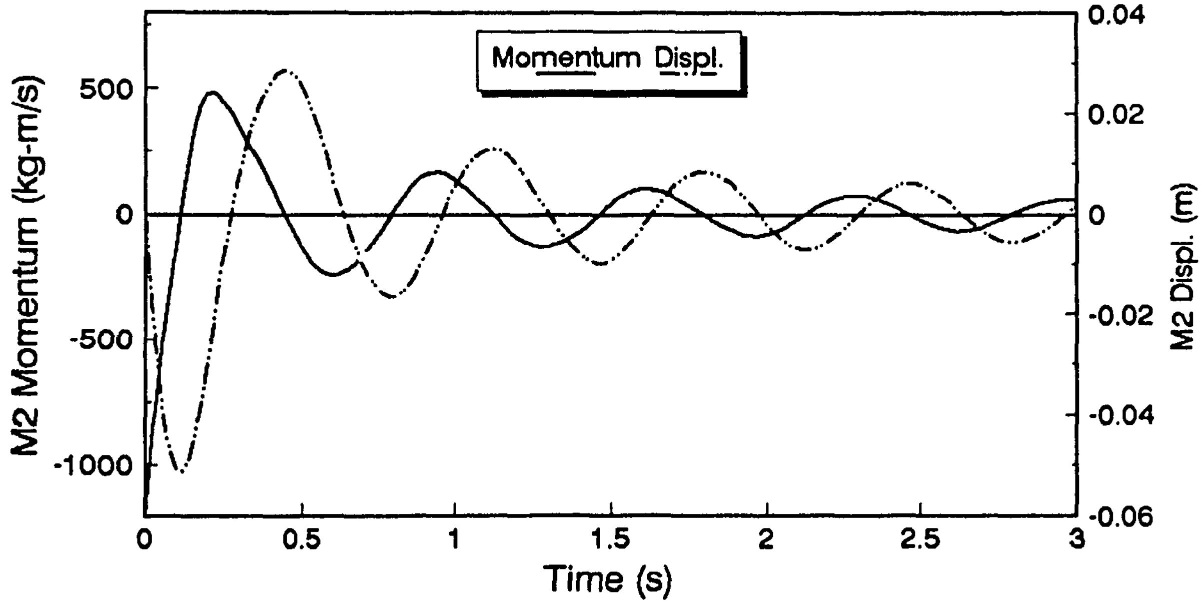


Fig. 4.30. Bounce Response of Linked-Cylinder Model

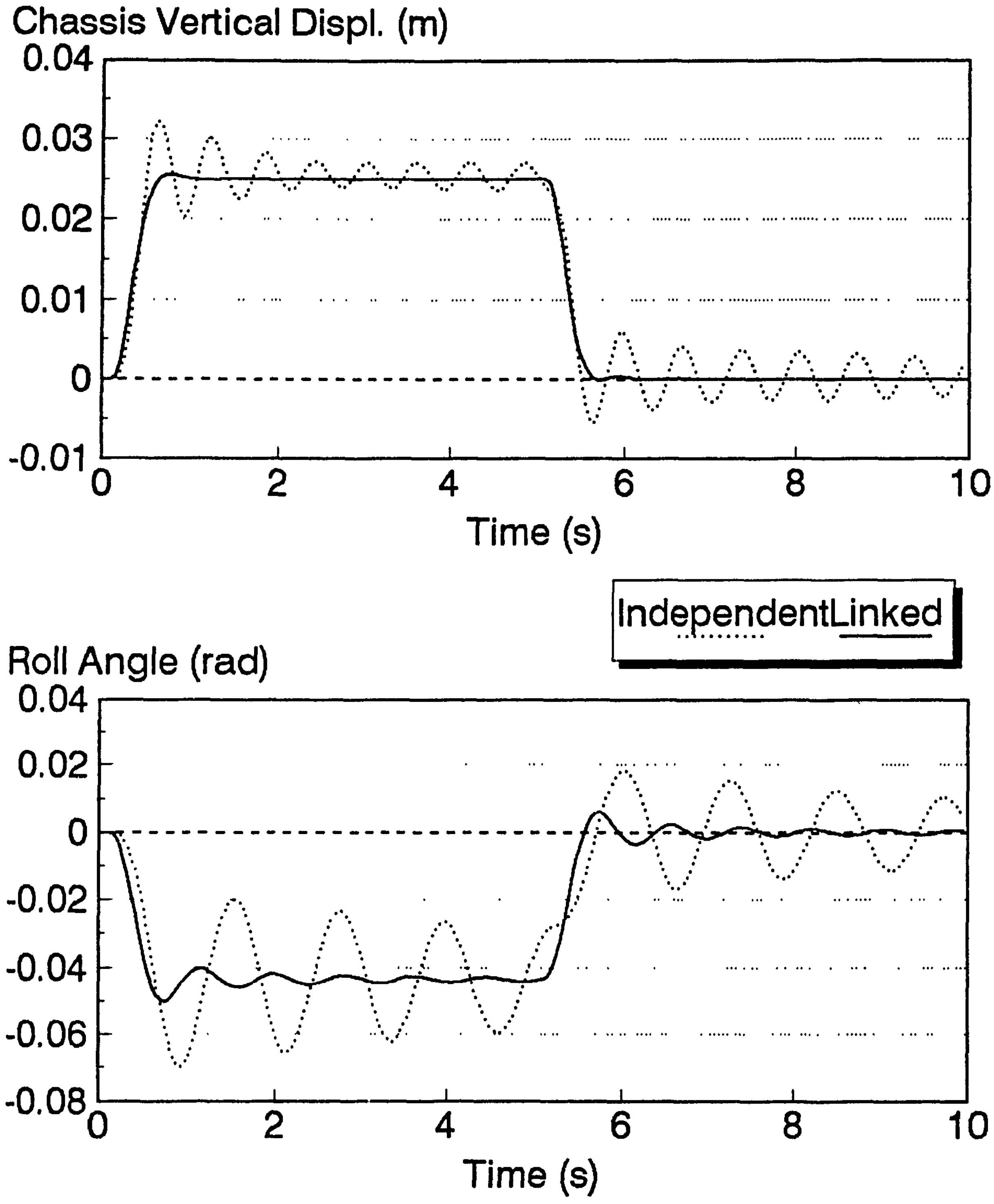


Fig. 4.31. Comparison of Independent and Linked-Cylinder Models to Asymmetrical Road Input

Other simulations (not shown) confirm the trend, showing that the smaller the area-ratio (meaning greater interconnection), the greater the reduction in transient roll response.

Series III: Cornering Behaviour

Cornering behaviour is simulated by assuming an applied moment, as already discussed. The third series of simulations (Fig. 4.32) shows that

- varying the area ratio is a useful tuning parameter for the roll dynamics
- interconnection of cylinders reduces steady-state cornering roll response, additionally to reducing transient road-induced roll response.
- the slight rise in chassis c.g. can best be unsderstood by considering the nonlinearity of the gas springs. One compresses less than the other expands.

The results obtained for this example, using the two suspension models agree qualitatively with the results published in [10]. Further confidence in the CANVAS modelling approach is obtained from the agreement of the numerically computed steady-state results with the analytical prediction.

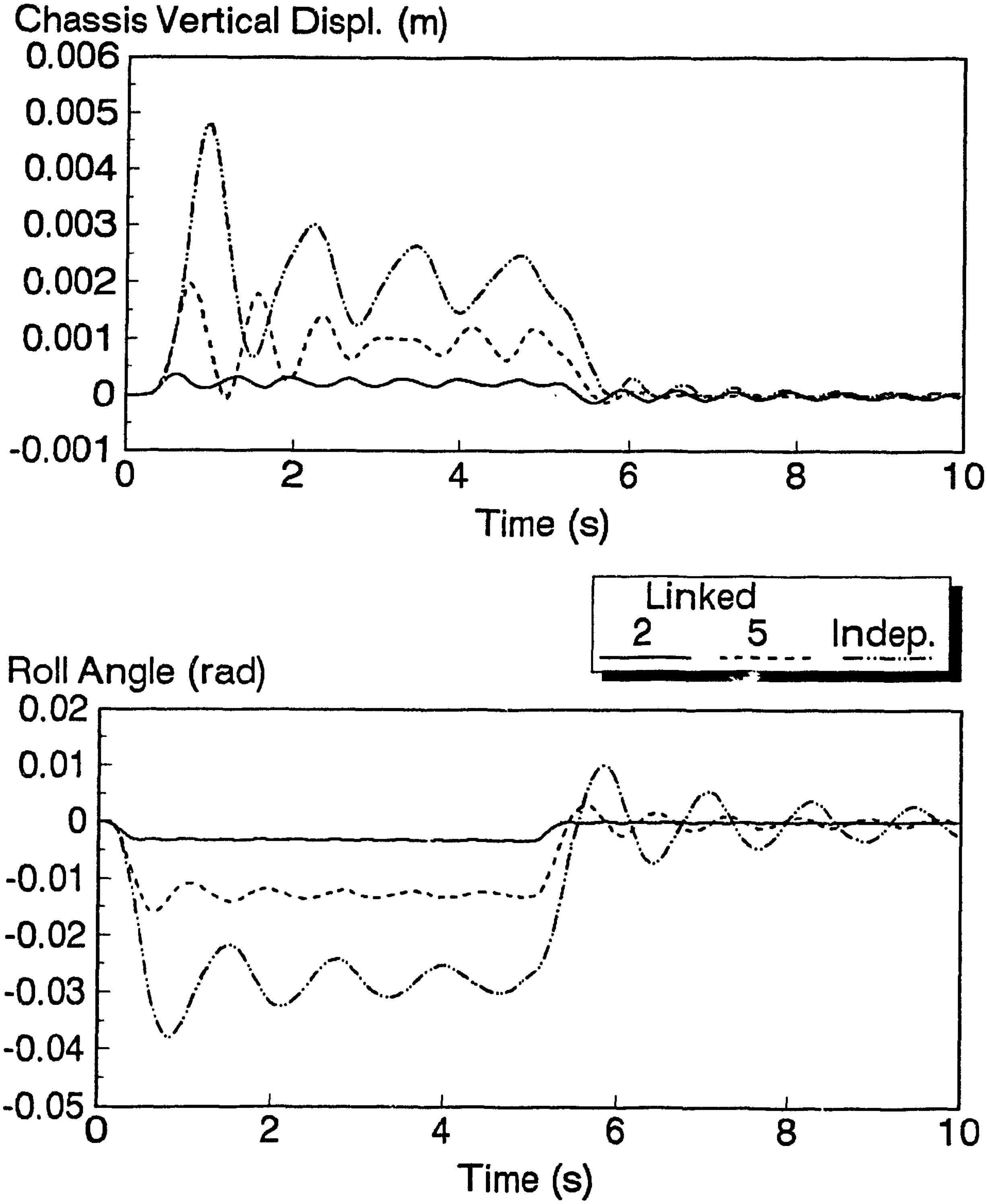


Fig. 4.32. Comparison of Cornering Response of Independent vs. Linked Cylinder Suspension for Two Area Ratios

4.3.3 Linear Optimal Active Suspension (THOMP155)

OBJECTIVE

The examples shown so far illustrate the use of built-in components provided by CANVAS. Because the use of a closed set of very specific component models is found too restrictive, the software includes some black-box elements used to 'bridge the gap' with the capabilities of other systems. One such element is the arbitrary force generator (FGN), which we illustrate here.

REMARKS

The active suspension problem has been treated formally by several authors from the viewpoint of a linear optimal control problem, with a state-variable model of the vehicle. In particular, we wish to show how the CANVAS package may be used to generate the state variable model and to predict the dynamic response of a system including an active suspension of the form proposed by Thompson [45,46].

The schematic of the system is shown in Fig. 4.33 which we see is a direct equivalent of that appearing in [45] (reproduced as Fig. 4.34). The set of dynamical equations produced by CANVAS for this model is shown in Fig. 4.35. The force term (e1) is missing: it is desired to produce this by state variable feedback.

Using the comments appearing in the incomplete equation set, plus the state-table produced automatically by BOND_BAS, it is easy for the user of CANVAS to 'complete'

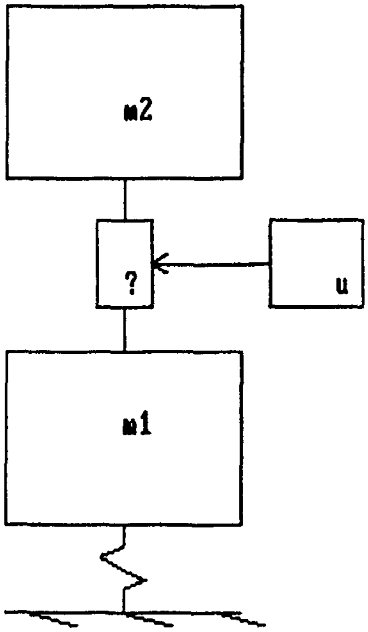


Fig. 4.33. CANVAS Schematic for 2 DOF Optimal Linear Active Suspension

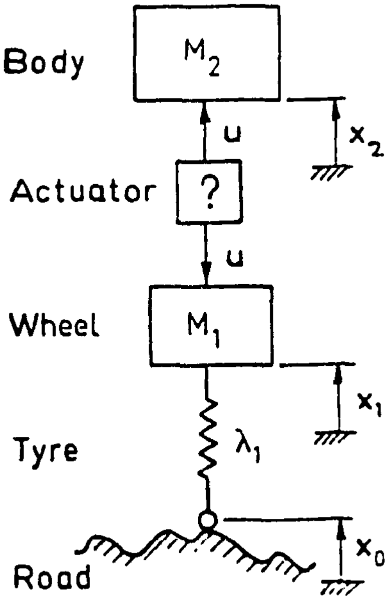


Fig. 4.34. Schematic of 2 DOF Optimal Linear Active Suspension (from [45])

File = Thom.INI

```
'thom
CONST F$ = "thom"
nState = 5
nOutputs = 1
nXCT = 0
DIM xctMean$(nXCT)
Z0(1) = 0: M$(1) = "body momentum"
Z0(2) = 0: M$(2) = "wheel momentum"
Z0(3) = 0: M$(3) = "spa displ."
Z0(4) = 0: M$(4) = "body disp"
Z0(5) = 0: M$(5) = "wheel disp"
M$(6) = "actuator force"
```

File = Thom.Out

```
f5 = z(2) / 28.58                         'wheel   I 6
s4 = f5                                   'wheel   1 7
Zdot(5) = s4
f3 = z(1) / 288.9                         'body   I 3
s2 = f3                                   'body   1 4
Zdot(4) = s2
f13 = f5                                  'wheel   1 7
s6 = 0                                    'grd   KON 9
f14 = (s6)                                'grd   SF 10
f10 = (f14 - f13)                         '-  0 14
Zdot(3) = f10                             '-  C 15
e10 = z(3) / 6.414E-06                    '-  C 15
e13 = e10                                 '-  0 14
e1 =                                      'fgn   SE 1
e12 = e1                                  'fgn   0 2
e5 = (-e12 + e13)                         'wheel   1 7
Zdot(2) = e5                              'wheel   I 6
e11 = e1                                  'fgn   0 2
e3 = (e11)                                'body   1 4
Zdot(1) = e3                              'body   I 3
X(6) = e1
'thom
```

Fig. 4.35. Incomplete State Equations, as Produced by CANVAS

the equations, and include any type of feedback. Of course, the feedback coefficients to be used are, at this stage, derived externally from the CANVAS system. It is possible, however, to integrate the capability for optimal control design into CANVAS at a later stage. For the present, we will utilize the feedback gains published in [45] which are reproduced in Table 4.2.

MODEL: THOMP155

Component	Parameters	Units	Represents
M1	m =28.58	kg	Unsprung mass
	$v_0=0$	m/s	
M2	m =288.9	kg	Sprung mass (body)
	$v_0=0$	m/s	
SPA	$C=6.414\times10^{-6}$	m/N	Tire (compliance)
GRD	-none-	m/s	Ground (zero vel.)
FGN	-none-	N	Optimal actuator

TABLE 4.2 - Feedback Gains Used in [45]

K_1	K_2	K_3	K_4
57240	-35355	1385.7	-4827.0

RESULTS

The completed set of equations is shown in Fig. 4.36. The response to a unit step displacement input (equivalent to a unit initial displacement) is plotted in Fig. 4.37 which shows, as expected, very close agreement to the published results.

File = Thom.INI

```
'thom
CONST F$ = "thom"
nState = 5
nOutputs = 1
nXCT = 0
DIM xctMean$(nXCT)
Z0(1) = 0: M$(1) = "body momentum"
Z0(2) = 0: M$(2) = "wheel momentum"
Z0(3) = 0: M$(3) = "spa displ."
Z0(4) = 0: M$(4) = "body disp"
Z0(5) = 0: M$(5) = "wheel disp"
M$(6) = "actuator force"
```

File = Thom1.Out

```
f5 = z(2) / 28.58                                 'wheel  I 6
s4 = f5                                           'wheel  1 7
Zdot(5) = s4
f3 = z(1) / 288.9                                 'body  I 3
s2 = f3                                           'body  1 4
Zdot(4) = s2
f13 = f5                                          'wheel  1 7
s6 = 0                                            'grd  KON 9
f14 = (s6)                                        'grd  SF 10
f10 = (f14 - f13)                                 '-  0 14
Zdot(3) = f10                                     '-  C 15
e10 = z(3) / 6.414E-06                            '-  C 15
e13 = e10                                         '-  0 14
'*********** STATE-FEEDBACK EQUATION
k1 = 57240: k2 = -35355: k3 = 1385.7: k4 = -4827
e1 = k1 * z(5) + k2 * z(4) + k3 * f5 + k4 * f3
'*********** ADDED BY DAN NEGRIN (90.2.5)
e12 = e1                                          'fgn  0 2
e5 = (-e12 + e13)                                 'wheel  1 7
Zdot(2) = e5                                      'wheel  I 6
e11 = e1                                          'fgn  0 2
e3 = (e11)                                        'body  1 4
Zdot(1) = e3                                      'body  I 3
X(6) = e1
'thom
```

Fig. 4.36. Completed State Equations, Including State-Feedback

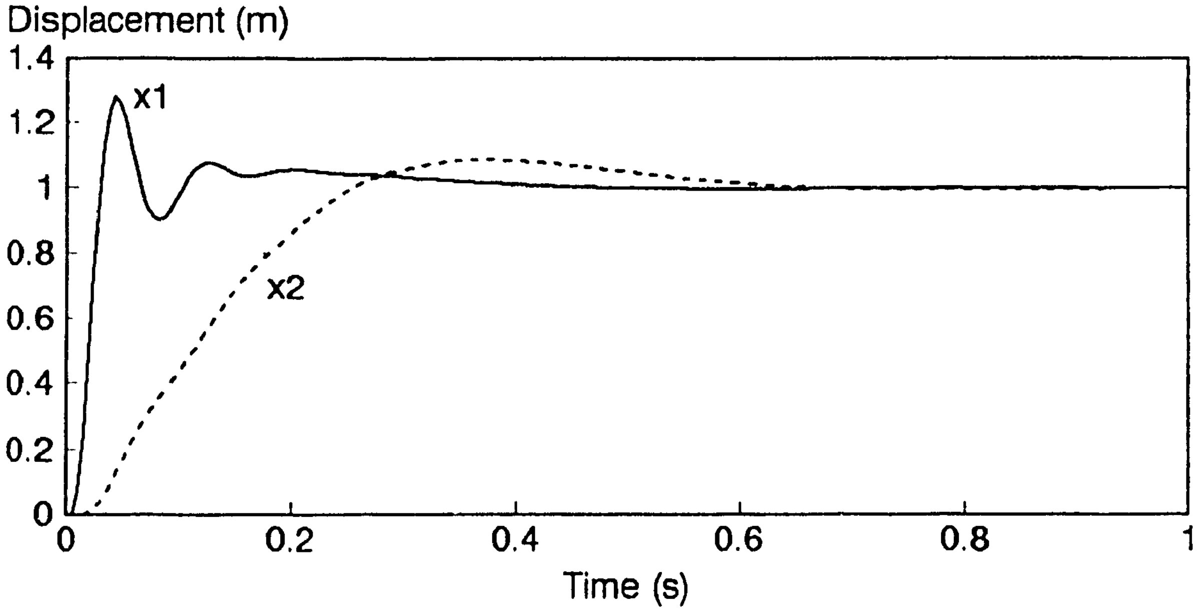


Fig. 4.37. Step Response of Optimal Linear Active Suspension

4.4 Summary

From a consideration of the sample models simulated using CANVAS, we see that the approach of schematic assembly yields valid models useful in evaluating the dynamic behaviour of a variety of physical systems.

The three-phase approach to validation has been applied to a number of models - some very simple, some more complex. Table 4.3 is a summary of the validation phases as they apply to the different examples.

It has been shown in this chapter that:

- The conversion of schematic to bond graph yields valid bond graph models.
- CANVAS solutions agree with analytical predictions.
- CANVAS is an attractive alternative to manual model derivation (it produces self-documenting models from graphical input).
- The ability to reproduce the work of other researchers from a schematic input means CANVAS is a time-saving package.

TABLE 4.3 - Validation of Examples

Example \ Phase	I bondgraph structure	II equations	III results
Radar Pedestal			✓
1 DOF Oscillator		✓	✓
Hydro-gas strut			✓
Open Loop Control			✓
Formula One		✓	✓
Crane Vehicle	✓		✓
Lin. Opt. Susp.		✓	✓

CHAPTER 5

Conclusions and Recommendations for Future Work

5.1 CONCLUSIONS

The software package CANVAS, which is the tangible result of the research work carried out for this thesis has shown that:

- arbitrarily complex physical system models can be built using a user-friendly graphical environment
- state-variable equations are automatically generated for nonlinear dynamical systems
- bond graphs can be applied to physical systems modelling as an intermediate language, while being fully transparent to the engineer
- results of several published papers may be reproduced with a savings in modelling effort, albeit with restrictions
- multi-domain physical systems can be modelled by a single software package while retaining the domain-dependent information

More specifically, we have shown that an engineer can construct a variety of physical system models using easy-to-understand graphical symbols (icons); the models are:

- numerically solvable
- symbolically represented by
  - i) bond graph
  - ii) equation file
- self-documenting

It was also demonstrated that bond graphs provide a useful intermediate notation between a physical system schematic and a symbolic equation form of a mathematical model. The full flexibility of the bond graph modelling approach is retained within CANVAS; however, the user has access to a higher level description of the model, based on graphical icons representative of the physical components in the actual system. Thus, we have not only the virtual capability of the general-purpose bond graph package (viz. ENPORT-7), but also a user-friendly and domain-specific graphical front-end to help with the creation of system models.

Distinguishing CANVAS from other self-contained analysis packages (such as ENPORT-7) is the openness of the modelling. The user is free to modify the derived model equations (using any text editor) and to incorporate them into any program of his/her choice for analysis.

Several example models were developed for both validation and application purposes. A first series of examples was constructed to show the manner in which the software, consisting of several modules, was verified. The second series of examples served to illustrate the manner in which CANVAS forms a viable alternative to the traditional, manual approach to system modelling.

The inclusion of pre-existing component models from several physical domains (hydraulic, mechanical and control) was made possible by using bond graphs as the underlying representation.

5.2 RECOMMENDATIONS FOR FUTURE WORK

It is clear that the current implementation of CANVAS is far from complete. Several of the restrictions that were imposed in order to limit the scope of the project can be removed, thus enhancing the capabilities of the package. This section suggests some of the more important issues to be addressed in future additions to CANVAS.

Non-Integral Causality

The current requirement for full integral causality is a matter of mere computational convenience. In a future version we will treat implicit equations as a separate case, since they are much more computationally demanding than explicit equations, requiring iterative solution of algebraic constraints at every time step of the integration.

Component Connection Models

As mentioned in Chapters 3 and 4, connections made by the user within the preprocessor are treated in the simplest possible way by CANVAS. A later enhancement to the software should allow for modelling of connections in the following additional ways:

- mechanical system constraint (e.g. revolute joint)
- dynamic model for hydraulic line

RVJ Revolute Joint

The connection of mechanical inertias through constraint elements is not treated in the current implementation of CANVAS. Constraints such as revolute joints arise often in mechanical systems, and they serve to reduce the number of degrees of freedom. Some software packages augment the basic set of differential equations with algebraic constraint equations, thus increasing the complexity of the ensuing model.

On the other hand it is possible, through a topological analysis of the mechanical portion of a system model, to actually eliminate degrees-of-freedom from a model when constraints are used. This improves the efficiency of the resulting equations, and has been done in the package CAMSYD [37].

It is also possible to work directly on the bond graph structure to effectively construct pre-reduced models when constraints are introduced into the schematic. This is because the bond graph is simply a representation of the mathematical equations. It is analogous to the bond-graph pattern simplification already discussed in Chapter 3.

LIN Hydraulic Line

The current implementation of CANVAS models all hydraulic lines using simple power bonds. This is equivalent to a lossless line, in which any resistive or capacitive effect is considered negligible. A future refinement will provide for conditional inclusion of line capacitance, resistance or inertance effects. These effects are sometimes important, as Dransfield [4] discusses:

> In most hydraulic control systems, the lines are short and of reasonably large diameter to keep fluid flow velocities to a low value...There may be cases however, where it is desirable to include some or all of a line's R,C, and I effects in a dynamic model. The situation could arise if
>
> - the lines are long or of small bore
> - the system is being driven in a highly dynamic mode

Felez and Vera [10] also consider the line model to be of importance. They find that by modelling the hydraulic line as a delay, the control system can easily be made unstable.

Additional Components

The set of components developed with CANVAS, although extensive, is limited to relatively primitive 'blocks'. More numerous and more complex component models are necessary to extend the applicability of the package.

Some useful mechanical components include: three degree-of-freedom rigid body, for bounce pitch and roll; flexible body element; revolute joint; gears; etc.

More hydraulic system components are also needed. Some of the more useful include realistic pump and motor models; more valves; the aforementioned hydraulic line.

For completeness, more transducers are also needed, for such dynamic variables as pressure and flow. Also, transducer dynamic effects may be of interest.

Open-endedness

The main limitation in model construction and evaluation within CANVAS arises from having a closed set of pre-defined component models from which to choose. Expandability of the modelling framework is a must if the package is to be truly general.

There are 'hooks' in the software for the construction of models that include arbitrary components. Examples of this are the arbitrary force generator block (FGN), the motion source (SMO) and the arbitrary excitation block (XCT). They allow the user to manually enter equations into the model.

A preferable mode of operation is to have a component model editor with which an engineer may create a graphical description of a new component (icon). Following this, a mathematical description of its behaviour, iı. the form of bond graph structure would be given by the user. Finally, port and connection information would be provided in order to interface to other, existing components. This is what is meant by open-endedness.

As a more limited implementation of this idea, it is possible for commonly used assemblies of components may be saved as modules that can be recalled as a unit for insertion in the circuit (standard subsystems).

Frequency Response

The FFT feature in the POST module provides a way to compute frequency-dependent response information for a model. However, a more general method should be sought, so that CANVAS can be used to automatically derive frequency response functions for its models. It must be noted that the linearity of a CANVAS model depends on the linearity of its component models. Thus, it is known *a-priori* if a model is linear, and therefore whether or not a transfer function can be written. Even with nonlinear components present, an option can be included to use a linearized form of the constitutive equations.

Numerical Integration

The simple numerical integration method used to solve state equations in CANVAS can be replaced with a more appropriate numerical integrator. The ability to model mixed-domain system often leads to models with widely-spaced natural modes, requiring the adoption of an appropriate adaptive stepsize integrator with error control.

Further Validation

Software validation as pursued in this thesis can be supplemented by comparison of CANVAS w th other packages such as MEDYNA [115] and HYSAN [29]. This would also provide an understanding of the relative merits and pitfalls of our approach.

Module Integration

The transformation of a schematic to a bond graph is, in the current implementation of CANVAS, performed after the schematic is completed. Thus, causal conflicts and schematic-derived inconsistencies are detected outside of the interactive schematic creation step. It would be highly desirable to implement an incremental bond graph creation and equation writing scheme to provide the user with direct feedback on the model development.

The value of this cannot be overemphasized, since one of the greatest advantages of bond graphs, causal analysis, is potentially available to the user as guidance in the model construction. If causal conflicts are detected during schematic construction, the user can be advised to alter the model in a timely and 'expert' fashion.

Knowldege-Based Modelling

The modular representation of system models, augmented with compatibility infcrmation and parameters forms a framework upon which a knowledge-based modelling system may be built. Such a knowledge-based system could automatically select appropriate model entities when sufficient information is presented to it, thereby ensuring that a model is a reasonable representation of the corresponding physical system.

Specifically, consider the case of modelling a vehicle's ride dynamics. Choices that face the modeller include the number of degrees of freedom, the possible inclusion of damping at strategic points in the vehicle (tire, bushings). Other examples may be the eventual necessity to include fluid compliance effects in hydraulic portions of the model, or frictional/leakage losses in actuators, valves and the like.

As each model is expected to yield forth a specific set of answers to a limited number of questions, it is reasonable to project that a modelling knowledge base can be built from current expertise to assist the modeller in making choices of element types and/or constitutive element laws under difficult circumstances. Such a knowledge base would consist of a set of associations between situations and actions (production rules [47]).

Learning

The flexibility of the knowledge-based system would greatly be enhanced by providing it with the capacity to learn. This means that in the case where the system fails to produce an answer (when presented with an unfamiliar case), it would ask the user a series of questions aimed at building more rules to help it arrive at a valid conclusion. Such 'learning' systems exist [48], and their implementation is not an overwhelmingly complicated task. Sufficient for its creation is a simple representation of production rules wich is not 'compiled', but rather interpreted.

REFERENCES

1. *ACSL, User-Guide/Reference Manual*, Third Edition, Mitchell and Gauthier Associates, Concord, MA, 1981.

2. Rosenberg, R.C. and Karnopp, D.C., Introduction to Physical System Dynamics, McGraw-Hill, New York, 1983.

3. Paynter, HM, Analysis and Design of Engineering Systems, MIT Press, Cambridge, MA, 1961.

4. Dransfield, P., Hydraulic Control Systems - Design and Analysis of Their Dynamics, Springer-Verlag, New York, 1981.

5. Pacejka, H.B., "Modelling Complex Vehicle Systems Using Bond Graphs", J. Franklin Inst., Vol. 319, No. 1/2, pp. 67-81, 1985.

6. Karnopp, D.C. and Rosenberg, R.C., Analysis and Simulation of Multiport Systems - The Bond Graph Approach to Physical System Dynamics, MIT Press, Cambridge, Mass., 1968.

7. Tiernego, M.J.L. and Bos, A.M., "Modelling the Dynamics and Kinematics of Mechanical Systems with Multibond Graphs", J. Franklin Inst., Vol. 319, No. 1/2, pp. 37-50, 1985.

8. Barreto, J. and Lefevre, J., "R-Fields in the Solution of Implicit Equations", J. Franklin Inst., Vol. 319, No. 1/2, pp. 227-236, 1985.

9. Hood, S.J., Rosenberg, R.C., Withers, D.H. and Zhou, T., "An Algorithm for Automatic Identification of R-Fields in Bond Graphs," IBM J. Res. Develop., Vol. 31, No. 3, pp. 382-390, 1987.

10. Felez, J. and Vera, C., "Bond Graph Assisted Models for Hydro-Pneumatic Suspensions in Crane Vehicles", Veh. Syst. Dyn., Vol. 16, pp. 313-332, 1987.

11. Karnopp, D.C., "Bond Graphs for Vehicle Dynamics", Veh. Syst. Dyn., Vol. 5, pp. 171-174, 1976.

12. Hubbard, M. and Karnopp, D., "Modelling of Vehicle Bump and Skid Response Using Bond Graphs", Int. J. of Vehicle Design, Vol. 4, No. 5, pp. 511-523, 1983.

13. Bos, A.M., "A Bond Graph Approach to the Modelling of a Motorcycle", Proc. 3rd ICTS Seminar, Amalfi, Italy, May 5-10, pp 289-314, 1986.

14. Pacejka, H.B. and Tol, C.G.M., "A Bond Graph Computer Model to Simulate the 3D Dynamic Behavior of a Heavy Truck", Proc. 10th IMACS World Congress *System Simulation and Scientific Computation*, Montreal, 1982 (Ed. by W.F. Ames et al), pp. 161-165, North-Holland, 1983.

15. Margolis, D.L., "A Survey of Bond Graph Modelling for Interacting Lumped and Distributed Systems", J. Franklin Inst., Vol. 319, No. 1/2, pp. 125-135, 1985.

16. Samanta, B., Mukherjee, A. and Deb, K., "Bond Graph Adapted Modular Approach to Analysis of Planar Mechanisms", Proc. 7th World Congress, *The Theory of Machines and Mechanisms*, Sevilla, Spain, 17-22 Sep. 1987, (Ed. by J. Bautista et al), Vol. 1, pp. 439-442, Pergamon Press, Oxford, 1987.

17. Zeid, A., "Bond Graph Modeling of Planar Mechanisms with Realistic Joint Effects", J. Dyn. Syst. Meas. and Control, Vol. 111, pp. 14-24, Mar. 1989.

18. Fahrenthold, E.P. and Wu, A., "Bond Graph Modeling of Continuous Solids in Finite Strain Elastic-Plastic Deformation", J. Dyn. Syst. Meas. and Control, Vol. 110, pp. 284-287, Sep. 1988.

19. Grabowiecka, A. and Grabowiecki, K. A., "Application of Bond Graphs to the Digital Simulation of a Two-Stage Relief Valve Dynamic Behaviour", Proc. IFAC Symp., *Pneumatic and Hydraulic Components and Instruments in Automatic Control*, Warsaw, May 22-23, 1980, Edited by H.J. Leskewicz and M. Zaremba, Pergamon Press, Oxford, pp. 15-20, 1980.

20. Meerman, J.W., "THTSIM, Software for the Simulation of Continuous Dynamic Systems on Small and Very Small Computer Systems", Int. J. Modelling and Simulation, Vol. 1, No. 1, pp. 52-56, 1981.

21. Beukeboom, J.J.A.J., Van Dixhoorn, J.J. and Meerman, J.W., "Simulation of Mixed Bond Graph and Block Diagram on Personal Computers Using TUTSIM", J. Franklin Inst., Vol. 319, No. 1/2, pp. 37-50, 1985.

22. *The Enport Reference Manual*, ROSENCODE Associates, Inc., Lansing, MI, 1987.

23. Granda, J.J., "Computer Generation of Physical System Differential Equations Using Bond Graphs", J. Franklin Inst., Vol. 319, No. 1/2, pp. 243-255, 1985.

24. Lorenz, F. and Wolper, J., "Assigning Causality in the Case of Algebraic Loops", J. Franklin Inst., Vol. 319, No. 1/2, pp. 237-241, 1985.

25. Szewczyk, K., Sobczyk, A., Rostkowski, A., Rybicki, A. and Zgorzelski, M., "On an Example of Comparing Experimental Investigations and Digital Simulation of a Hydraulic Installation, Using a PRBS Generator", *Proc. IFAC Symp., Pneumatic and Hydraulic Components and Instruments in Automatic Control,* Warsaw, May 22-23, 1980, Edited by H.J. Leskewicz and M. Zaremba, Pergamon Press, Oxford, pp. 33-37, 1980.

26. Vilenius, M.J, et al, "CATSIM - A New Kind of Computer Program for Hydraulic Circuit Design", Proc. 7th Int. Fluid Power Symp., Bath, England, Sep. 1986.

27. Krus, P. and Palmberg, J.O., "Simulation of Fluid Power Systems in Time and Frequency Domains", Proc. 7th Int. Fluid Power Symp., Bath, England, Sep. 1986.

28. Kinoglu, et al, "Streamlining Hydraulic Circuit Designs With Computer Aid", *Computers in Mechanical Engineering,* Vol. 1, No. 2, pp 21-26, 1982.

29. *HYSAN: Software for Hydraulic System Analysis,* pamphlet, Hydrasoft Corp., Morton, IL, 1988.

30. *Advanced Fluid System Simulation,* Report No. AFAPL-TR-76-43, National Technical Information Service, April 1980.

31. Hargreaves, B., "GMR DYANA: The Computing System and its Applications", General Motors Engineering Journal, Vol. 8, No. 1, pp. 7-13, 1961.

32. Hargreaves, B., "GMR DYANA: Extending the Computing System to Solve More Complex Problems", General Motors Engineering Journal, Vol. 8, No. 1, pp. 14-18, 1961.

33. Dix, R.C. and Lehman, T.J., "Simulation of the Dynamics of Machinery", ASME J. of Engineering for Industry, Vol. 94, No. 2, pp. 433-438, 1972.

34. Maruyama, K. and Fujita, T., "Automatic Generation of Equations of Motion from Graphic Input of Vibration Model", JSME Int'l Journal, Series III, Vol. 31, No. 2, pp. 400-408, 1988.

35. Kortüm, W., "Modelling and Simulation of Actively Controlled Mechanical Systems", IMACS Transactions on Scientific Computation, 10th IMACS World Congress on Systems Simulation and Scientific Computation, Montreal, 8-13 Aug. 1982, North-Holland Publishing Co, 1983.

36. *DADS User's Manual, Version 5.0,* CADSI Inc., 1988.

37. Sankar, S., Alanoly, J. and Negrin, D., "Vehicle Ride Analysis Using Interactive Computer Graphics", Int. J. of Vehicle Design, Vol. 10, No. 3, pp. 347-367, 1989.

38. Thoma, J.U., Introduction to Bond Graphs and Their Applications, Pergamon Press, Oxford, 1975.

39. Horton, D.N.L. and Crolla, D.A., "Theoretical Analysis of a Semi-Active Suspension Fitted to an Off-Road Vehicle", Veh Syst Dyn, Vol. 15, pp. 351-372, 1986.

40. Hood, S.J., Palmer, E.R. and Dantzig, P.M., "A Fast, Complete Method for Automatically Assigning Causality to Bond Graphs", J Franklin Instn, Vol. 326, No. 1, pp. 83-92, 1989.

41. Margolis, D.L., "Bond Graphs as A Simulation Modelling Formalism", Proc. 3rd ICTS Seminar, Amalfi, Italy, pp. 233-262, 1986.

42. Schiehlen, W.O., "Vehicle Dynamics Application", NATO ASI series, Vol. F9, Springer-Verlag, Berlin, 1984.

43. Gear, CW, "Differential Algebraic Equations", NATO ASI series, Vol. F9, Springer-Verlag, Berlin, 1984.

44. Dominy, J. and Bulman, D.N., "An active suspension for a Formula One Grand Prix racing car", *ASME J. Dyn. Syst., Meas., and Control*, Vol. 107, pp 73-78, 1985.

45. Thompson, A.G., "An Active Suspension with Optimal linear State Feedback", Veh. System Dyn., Vol. 5, pp. 187-203, 1976.

46. Thompson, A.G., "Optimal and suboptimal linear suspensions for road vehicles", Veh. System Dyn., Vol. 13, pp. 61-72, 1984.

47. Hayes-Roth, F., Waterman, D.A. and Lenat, D.B. (eds), Building Expert Systems, Addison-Wesley, Reading, Mass., 1983.

48. Quinlan, J.R., "Semi-autonomous acquisition of pattern-based knowledge", in Introductory Readings in Expert Systems, Michie, D. (ed), Gordon and Breach, London, 1982.

APPENDIX 1

List of CANVAS Components and Elements

COMPONENTS

Mechanical

M1	Lumped Mass
M2	Two-degree-of-freedom rigid body
SPA	Linear Translational Spring
DMA	Linear Translational Damper
SD	Spring/Damper
GRV	Gravity
SMO	Motion source (velocity profile)
SEF	Force source

Hydraulic

ACC	Gas-charged accumulator
PMP	Fixed displacement pump
CY1	Single-acting cylinder
CY2	Double-acting cylinder
RLV	Relief valve
CV1	Proportional control valve
RSV	Constant-pressure reservoir

BONDGRAPH ELEMENTS

Source

SE	Source of Effort
SF	Source of Flow

Storage

I	Inertance
C	Capacitance (linear)
C:poly	Capacitance (polytropic)

Dissipation

R	Resistance (linear)
R:square	Resistance (square)
R:relief	Square resistance with threshold

Junction

0	Common-effort junction
1	Common-flow junction
TF	Transformer
GY	Gyrator
MTF	Modulated Transformer
MGY	Modulated Gyrator

BLOCK-DIAGRAM ELEMENTS

Function

ABS	Absolute Value
ATT	Attenuator
BKL	Backlash
COS	Cosine function
DEL	Delay (one simulation step)
DIV	Divides Input-a by Input-b
EXP	Exponential (base e)

FIX	Integer Value
GAI	Gain block
LIM	Limiter (saturation)
LOG	Natural logarithm
MAX	Maximum value amongst inputs
MIN	Minimum value amongst inputs
MUL	Product of inputs
PWR	Power
SGN	Signum of sum of inputs
SIN	Sine function
SQT	Square root
SUM	Sum inputs

Logic

AND	Logic AND of inputs
DFF	D-type flip-flop
INV	Logic inverter
NAN	NAND of inputs
NOR	NOR of inputs
ORR	OR of inputs
REL	Relay (select one of two inputs)
XOR	XOR of inputs

Memory

DIF	Derivative
EUL	Euler method integrator
FIO	First-order lag

INT	Default integrator RK4
IWZ	Integrator-With-Zero
LLG	Lead-Lag block
LME	Limited Euler integrator
LMI	Limited integrator
PID	PID controller
RIN	Resettable integrator
SEO	Laplace second-order block

Source

BMP	Single sinusoidal bump
CLK	Clock (pulsetrain)
CRP	Chirp (frequency sweep)
FRQ	Frequency source
KON	Constant value
PLS	Pulse
RMP	Ramp
RP2	Rounded pulse - given area
RPL	Rounded pulse - given amplitude
TIM	Time in simulation seconds

Special

HYS	Hysteresis
TXT	Text, used for labels in schematic